# SYLLABUS

**(With effect from 2013-2014)**
**Out Come Based Education**

# ಪಠ್ಯಕ್ರಮ

**(ಶೈಕ್ಷಣಿಕವರ್ಷ 2013–14)**
**ಫಲಿತಾಂಶ ಆಧಾರಿತ ಶಿಕ್ಷಣ**

## V and VI Semester

### Bachelor Degree
### in
### Information Science and Engineering

## P.E.S. College of Engineering

Mandya - 571 401, Karnataka
(An Autonomous Institution Affiliated to VTU, Belagavi)
Grant -in- Aid Institution
(Government of Karnataka)
Accredited by NBA, New Delhi
Approved by AICTE, New Delhi.

**ಪಿ.ಇ.ಎಸ್. ತಾಂತ್ರಿಕ ಮಹಾವಿದ್ಯಾಲಯ**
**ಮಂಡ್ಯ–571 401, ಕರ್ನಾಟಕ**
**(ವಿ.ಟಿ.ಯು, ಬೆಳಗಾವಿ ಅಡಿಯಲ್ಲಿನ ಸ್ವಾಯತ್ತ ಸಂಸ್ಥೆ)**

Ph : 08232- 220043, Fax : 08232 – 222075,Web : www.pescemandya.org

## Preface

PES College of Engineering, Mandya, started in the year 1962, has become autonomous in the academic year 2008-09. Since, then it has been doing the academic and examination activities successfully. The college is running eight undergraduate and eight Postgraduate programs. It consists of six M.Tech programs, which are affiliated to VTU. Other postgraduate programs are MBA and MCA.

India has recently become a Permanent Member by signing the Washington Accord. The accord was signed by the National Board of Accreditation (NBA) on behalf of India on 13th June 2014. It enables not only the mobility of our degree globally but also establishes equivalence to our degrees with that of the member nations such as Taiwan, Hong Kong, Ireland, Korea, Malaysia, New Zealand, Russia, Singapore, South Africa, Turkey, Australia, Canada and Japan. Among other signatories to the international agreement are the US and the UK. Implementation of Outcome Based Education (OBE), has been the core issue for enabling the equivalence and of Indian degrees and their mobility across the countries.

*Our Higher Educational Institution has adopted the semester structure with OBE scheme and grading system.*

The credit based OBE semester system provides flexibility in designing curriculum and assigning credits based on the course content and hours of teaching.

The OBE, emphasize setting clear standards for observable, measurable outcomes of programs in stages. There lies a shift in thinking, teaching and learning processes moving towards Students Centric from Teacher Centric education. OBE standards focus on mathematics, language, science, attitudes, social skills & moral values.

The key features which may be used to judge, if a system has implemented an outcome based education system is mainly Standard based assessments that determines whether students have achieved the stated standard. Assessments may take any form, so long as the process actually measure whether the student knows the required information or can perform the required task. Outcome based education is a commitment that all students of all groups will ultimately reach the same minimum standards. Outcome Based Education is a method or means which begins with the end in mind and constantly emphasizes continuous improvement.

In order to increase the Industry/Corporate readiness, many Soft Skills and Personality Development modules have been added to the existing curriculum of 2013-14.Industry Interactions have been made compulsory to enhance the field experience. In order to enhance creativity and innovation Mini Project is included in all undergraduate programs.

(Dr.H.V.RAVINDRA)
Dean (Academic)
Professor,
Dept. of Mechanical Engg.

(B.DINESH PRABHU)
Deputy Dean (Academic)
Associate Professor,
Dept. of Automobile Engg

**P.E.S.COLLEGE OF ENGINEERING, MANDYA-571401**

**Vision**

**"**An institution of high repute, imparting quality education to develop innovative and Humane engineers"

**Mission**

**"**Committed to develop students potential through high quality teaching- learning processes and state of the art infrastructure"

## DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

### About the Department

The Department of Information science and Engineering takes pride in producing quality engineers over the past 14 years. The credit for all the flowery results goes to the highly motivating staff, from whom all students draw inspiration. The Department was started in the year 2000. The present intake of the undergraduate program is 30. The department has well equipped classrooms, computer laboratories with high-end systems, department library and good collection of softwares. Also a research centre is a major credential to our department. We are proud to produce the first PhD student in our college. Faculty members of the department are involved in research activities in different fields such as Medical Image Processing, Pattern Recognition, and Data Mining etc. The department is using Outcome-based education (OBE), which is a recurring education reform model, and it is affiliated to Visvesvaraya Technological University (VTU). The department has achieved good Placement, conducted International /national Conferences and other sponsored short-term courses, workshops, National seminars and symposia. The laboratory facilities and the Internet access are available round the clock to the staff and students of the Information Science and Engineering.

- **VISION :**

"The department strives to equip our graduates with knowledge and expertise to contribute significantly to the Information Science technology industry".

- **MISSION:**

- To provide state-of-art facilities and to produce socially sensitive citizens.
- To prepare students for careers in industry and to peruse higher education.
- To promote leadership qualities among students.
- To encourage the faculty to pursue academic excellence through high quality research and publication.

**P.E.S.COLLEGE OF ENGINEERING, MANDYA-571401**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**(A) Programme Educational Objectives (PEOs)**

The Bachelor of Engineering Programme in Information Science and Engineering [B.E. (IS&E)] during four years term aims to

a) Establish a productive Information Science and Engineering career in industry, government or academia.
b) Engage in professional practice of information, computer and software systems engineering.
c) Promote the development of innovative systems and solutions using hardware and software integration.
d) Be successful in pursuing higher studies in engineering or management.
e) Pursue career paths in teaching or research.
f) Interact with their peers in other disciplines in industry and society and contribute to the economic growth of the country.

**(B) Programme Outcomes (POs):**

The BACHELOR OF ENGINEERING Programme in Information Science and Engineering [B.E. (IS&E)] must demonstrate that its graduates have

a) An ability to apply the knowledge of mathematics, science and computing appropriate to the Information Science and Engineering.
b) An ability to design and conduct experiments, as well as to analyze and interpret data.
c) An ability to design, implement and evaluate a computer-based system, component or program to meet desired needs.
d) An ability to function effectively on teams to accomplish a common goal.
e) An ability to identify, formulate and solve information and computer science engineering problems and define the computing requirements appropriate to their solutions.
f) An understanding of professional, ethical, legal, security and social issues and responsibilities.
g) An ability to communicate effectively with a range of audiences.
h) An ability to understand the impact of engineering solutions in a global, economic, environmental, and societal context; and to analyze the impact of computing on individuals, organizations, and society.
i) A recognition of the need for, and an ability to engage in life-long learning and continuing professional development.
j) Knowledge of contemporary Information Science and Engineering issues.
k) An ability to use current techniques, skills, and tools necessary for computing and engineering practice.
l) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of information and computer based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
m) An ability to apply design and development principles in the construction of software systems of varying complexity.

These programme outcomes (POs) are achieved through an array of courses. To ensure the achievement of POs, the course learning outcomes (CLOs) are so formulated that they address the POs.

# P.E.S.COLLEGE OF ENGINEERING, MANDYA-571401
## (An Autonomous Institution Under VTU. Belgaum)
## Department of Information Science & Engineering

V Semester B.E. (IS&E) SCHEME OF TEACHING AND EXAMINATION 2013- 14

| Sl. No. | Course Code | Course Title | Teaching Dept. | Hours/ Week L:T:P:H | Credits | Examination Marks | | | Exam Duration in hours |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | CIE | SEE | Total Marks | |
| 1. | P13IS51 | Unix System & Network Programming | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 2. | P13IS52 | Data Base Management System | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 3. | P13IS53 | Communication Networks | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 4. | P13IS54 | System Software | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 5. | P13IS55 | Software Engg. | IS&E | 2:2:0:4 | 3 | 50 | 50 | 100 | 3 |
| 6. | P13IS56 | Management and Entrepreneurship | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 7. | P13 ISL57 | System Software and Script lab | IS&E | 0:0:3:3 | 1.5 | 50 | 50 | 100 | 3 |
| 8. | P13ISL58 | Data Base Management System Lab | IS&E | 0:0:3:3 | 1.5 | 50 | 50 | 100 | 3 |
| 9 | P13HU59 | Professional and Efficient Avocation-I (PEA-I)* | HS&M | 2:0:0:2 | 0 | (50) | --- | --- | --- |
| 10 | P13ISL510 | Industry Interaction-II* | IS&E | 0:0:1:1 | 0 | (50) | -- | -- | --- |
| **Total** | | | | | 26 | 400 | 400 | 800 | |

**VI Semester B.E. . (IS&E) SCHEME OF TEACHING AND EXAMINATION 2013-14**

| Sl. No | Course Code | Course Title | Teaching Dept. | Hours/ Week L:T:P:H | Credits | Examination Marks | | | Exam Duration in hours |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | CIE | SEE | Total Marks | |
| 1. | P13IS61 | Object Oriented System Development | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 2. | P13IS62 | Computer Networks | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 3. | P13IS63 | Cloud Computing | IS&E | 2:2:0:4 | 3 | 50 | 50 | 100 | 3 |
| 4. | P13IS64 | C# and .NET | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 5. | P13IS65 | Modern Information Retrieval | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 6. | P13IS66* | Elective – I Group A | IS&E | 4:0:0:4 | 4 | 50 | 50 | 100 | 3 |
| 7. | P13ISL67 | C# and .NET Lab | IS&E | 0:0:3:3 | 1.5 | 50 | 50 | 100 | 3 |
| 8. | P13ISL68 | Networks Lab | IS&E | 0:0:3:3 | 1.5 | 50 | 50 | 100 | 3 |
| 9 | P13HU69 | Professional and Efficient Avocation-I (PEA-II)* | HS&M | 2:0:0:2 | 0 | (50) | --- | --- | |
| 10 | P13ISL610 | Mini Project-II* | IS&E | 0:0:1:1 | 0 | (50) | -- | -- | |
| **Total** | | | | | 26 | 400 | 400 | 800 | |

L**:** Lecture, T: Tutorial, P: Practical, CIE: Continuous Internal Evaluation, SEE: Semester End Examination, C: Credits **HC**: Hard Core (4Credits) -4 Courses **OS**: Other subject (3 Credits) – 1 Course **PS**: Professional subject (4Credirs)-1Course. **One Hour Lecture= Two Hours Tutorial / Practical = 1 Credit.**

* PEA-I, , Industry visit and interaction-II, PEA-II, Mini Project: All students shall have to pass this mandatory learning courses before completion of VIII-Semester

**** List of Electives-I (Group-A)**

| Sl.No | Course Code | Elective Course Title |
|---|---|---|
| 1. | P13IS 661 | Storage Area Networks |
| 2. | P13IS 662 | J2EE & J2ME |
| 3. | P13IS 663 | Artificial Intelligence & Agent Technology |
| 4. | P13IS664 | Multimedia Computing |
| 5. | P13IS665 | Principles of Compiler Design |

| **EVALUATION SCHEME** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Scheme** | **Weightage** | **Marks** | **Event Break Up** | | | | |
| **CIE** | 50% | 50 | **Test I** | **Test II** | **Quiz I** | **Quiz II** | **Assignment** |
| | | | 35 | 35 | 5 | 5 | 10 |
| **SEE** | 50% | 100 | **Questions to Set: 10** | | **Questions to Answer: 5** | | |
| **Scheme of SEE Question Paper (100 Marks)** | | | | | | | |
| **Duration: 3Hrs** | | | **Marks: 100** | | **Weightage: 50%** | | |

- Each of the two questions set shall be so comprehensive as to cover the entire contents of the unit.
- There will be direct choice between the two questions within each Unit
- Total questions to be set are 10. All carry equal marks of 20
- The number of subdivisions in each main question shall be limited to three only
- Number of questions to be answered by students is 5

| Course Title: UNIX System and Network Programming | | | |
|---|---|---|---|
| **Course Code: P13IS51** | **Semester : V** | **L- T – P-H : 4 – 0 – 0-4** | **Credit: 4** |
| **Contact period : Lecture: 52 Hrs, Exam:3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites :** Computer Concepts and C Programming and Object Oriented Programming with Java

## Course Learning Objectives (CLOs)

**This course aims to**

1. Explain the need and use of UNIX, ANSI and POSIX Standards.
2. Explain File APIs with an example.
3. Write a C/C++ program using File APIs.
4. Explain the UNIX kernal support for files and processes.
5. Write a C/C++ program using APIs available for handling Process.
6. Write a program in C/C++ to illustrate a race condition.
7. Explain how UNIX operating system keeps process accounting.
8. Explain POSIX defined signals with an example.
9. Write a C/C++ program to transform a normal user process into a daemon process.
10. Write a C program that sends a message to the child process through the different types of pipes.
11. List and explain the APIs used to create and control the semaphores, shared library and Message Queues.
12. Write a C/C++ program to illustrate the use of a socket.

## Course Content

### Unit – I

**Introduction:** UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.

**Unix Files:** File Types, The UNIX and POSIX File System, The UNIX and POSIX File Attributes, Inodes in UNIX System V, Application Program Interface to Files, UNIX Kernel Support for Files, Relationship of C Stream Pointers and File Descriptors, Directory Files, Hard and Symbolic Links.

**10 Hrs**

### Unit – II

**Unix File APIs:** General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs

**Unix Processes:** The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions. **10 Hrs**

### Unit – III

**Process Control:** Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, waited, wait3, wait4 Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times.

**Process Relationships**: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp, tcsetpgrp, and tcgetsid Functions, Job Control, Shell Execution of Programs, Orphaned Process Groups. **10 Hrs**

### Unit – IV

**Signals And Daemon Processes:** Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers.

**Daemon Processes:** Introduction, Daemon Characteristics, Coding Rules, Error Logging, Single instance daemons; Daemon conventions; Client-Server Model. 1**0 Hrs**

**Unit – V**

**Interprocess Communication:** Introduction; Pipes, popen, pclose Functions; Coprocesses; FIFOs; System V IPC; Message Queues; Semaphores, Shared Memory.
**Network IPC:** Sockets Introduction; Socket Descriptors; Addressing; Connection establishment; Data transfer; Socket options; Out-of-band data; Nonblocking and asynchronous I/O. **12 Hrs**

**Text Books:**

1. Terrence Chan: Unix System Programming Using C++, Prentice-Hall of India / Pearson Education, 2012. (Chapters 1, 5, 6, 7, 8, 9)
2. W.Richard Stevens, Stephen A. Rago: Advanced Programming in the UNIX Environment, Revised Edition, Pearson Education / Prentice-Hall of India. (Chapters 7, 8, 9, 13, 15, 16)

**Reference Books:**

1. Marc J. Rochkind Advanced Unix Programming, Revised Edition, Pearson Education.
2. Maurice.J.Bach The Design of the UNIX Operating System, Pearson Education / PHI.
3. Uresh Vahalia UNIX Internals, Pearson Education, 2013.

## Course Outcomes

**After learning all the units of the course, the student is able to**

1. Distinguish between ANSI C, K&R C and POSIX Standards
2. Describe different UNIX Files
3. Describe different UNIX APIs
4. Apply different UNIX APIs depending on the requirement
5. Implement modules of real world application using appropriate UNIX APIs

## Topic Learning Objectives

**After learning all the topics of unit – I, the student is able to**

1. Explain the need of standards – L2.
2. Distinguish between ANSI C and K&R C – L3.
3. Explain different subsets of POSIX standard – L2.
4. List and explain POSIX API's – L1, L2.
5. Calling an API is more time consuming than calling a user function. Justify or Contradict – L3.
6. List and explain different types of UNIX files – L1, L2.
7. List all the file attributes along with their meanings – L1, L2.
8. Describe the UNIX Kernel support for files – L2.
9. Distinguish between C stream pointers and file descriptors – L3.
10. Distinguish between Hard and Symbolic links – L3.

**After learning all the topics of unit – II, the student is able to**

1. ist and explain any two general file API's – L1, L2.
2. Explain the use of following APIs:
   i) read ii) write iii) close - L2.
3. Explain how fcntl API is used for file and record locking – L2.
4. With suitable examples explain various directory file APIs – L2.
5. Differentiate between hard link and symbolic link files with an example –L3.
6. Write the prototype of the main( ) function and explain the same –L2.
7. Explain the different exec functions and also explain how their functioning differs from each other - L2.
8. Explain briefly the memory layout of a C program – L2.
9. With an example explain the use of setjmp and longjmp functions – L2.
10. With an example explain the use of getrlimit and setrlimit functions – L2.

**After learning all the topics of unit – III, the student is able to**

1. What is fork and vfork ? Explain the same with an example – L1, L2.
2. Explain the following functions with an example:
   i) wait ii) wait4 iii) waited - L2.
3. Define race condition. Write a program to demonstrate the race condition – L1, L3.
4. Explain how accounting is done in UNIX system – L2.
5. Explain the need of process timers – L2.
6. Distinguish between Terminal login and Network Login –L3.
7. Define a session – L1.
8. Explain the use of tcgetpgrp – L2.
9. Explain tcsetpgrp and tcgetsid functions – L2.
10. Explain shell execution of programs – L2.

**After learning all the topics of unit – IV, the student is able to**

1. Define a signal – L1.
2. Explain the UNIX kernal support for signals – L2.
3. Define a Signal Mask – L1.
4. Explain the prototype of waitpid function – L2.
5. Explain sigsetjmp and siglongjmp functions with an example – L2.
6. List the timer manipulation APIs in POSIX.1b – L1.
7. Define daemon processes. Enlist their characteristics. – L1, L2.
8. Explain the need of error logging – L2.
9. Explain Daemon conventions – L2.
10. Explain Client-Server model – L2.

**After learning all the topics of unit – V, the student is able to**

1. Define Inter Process Communication – L1.
2. Write a c program that sends "hello world" message to the child process through the pipe. The child on receiving this message should display it on the standard output – L3.
3. Write a c program that sends "hello world" message to the child process through the pipe. The child on receiving this message should display it on the standard output using popen and pclose functions – L3.
4. Explain Coprocesses with an example – L2.
5. Explain how FIFO can be used to implement client-server communication model – L2.
6. List and explain the APIs used to create and control the Message Queues – L1, L2.
7. List and explain the APIs used to create and control the semaphores – L1, L2.
8. Explain Socket Descriptors – L2.
9. Explain Connection establishment with an example – L2.
10. Explain the need of asynchronous I/O – L2.

**Review Questions**

1 What is posix standard? Explain the different subset of posix standard.
2 Explain the common characteristics of API and describe the error status codes.
3 List the differences between ANSI C and K & R C.Explain.
4 Explain any five error status code for error no.
5 Write a c or c++posix complement program to check following limits:
   i)Number of clock ticks
   ii)Maximum number of child processes
   iii)Maximum path length
6 Explain POSIX Feature Test Macros with examples.
7 Explain the different file types available in UNIX or POSIX system.

8 Describe the UNIX Kernel support for files.
9 What is an API? How is it different from c library functions?
10 What are APIs? When do you use them? Why are the APIs more time consuming than the library function?
11 List all the file attributes along with their meanings. Which of these attributes can't be changed and why?
List the commands needed to change the following file attributes:
i) filesize;ii) User ID; iii) Last access and modification time; iv) hard link count.
12 What is an inode? Why are the inodes unique only within a file system? How does OS map the inode to its filename? Bring out the four important differences between soft and hardlinks.
13 List and explain the access mode flags and access modifier flags. Also explain how the permission value specified in an „open" call is modified by its calling process "umask" value.
14 Explain how fcntl API is used for file and record locking.
15 Write a program to implement ls –l command.
16 List the structure used to query the file attributes in Unix. Write a program in c++ to list the following file attributes of a given regular file passed as command line argument i) file type ii)hard link count iii)file size iv)file name
17 Write a c++ program tocheck whether the close – on – exec flag is set for a given file. If it is not set, use fcntl to set thisflag. Also show the implementation of dups macro using this API.
18 List the important uses of fcnt API. Give its prototype description.
19 Write an explanatory note on environment variables. Also write a C/C++ program that Outputs the contents of its environment list.
20 With an example explain the use of setjmp and longjmp functions.
21 Describe the UNIX Kernel support for process. Show the related data structures.
22 Bring out the importance of locking files. What is the drawback of advisory lock? Explain in brief.
23 What are the different ways in which a process can terminate? With a neat block schematic, explain how a process is launched and terminates clearly indicating the role of C-startup routine and the exit handlers.
24 With a neat diagram, explain the memory layout of c program. In which segments are the automatic variables and dynamically created objects are stored?
25 Explain the following system calls: i)fork ii)vfork iii)exit iv)wait.
26 Giving the prototype explain different variant of exec system call.
27 What is race condition? Write a program in C/C++ to illustrate a race condition.
28 How UNIX operating system keeps process accounting?
29 What is job control? Summarize the job control features with the help of a figure.
30 Explain the following:i)wait ii)waitpid
31 What is a signal? Discuss any five POSIX defined signals.
32 What is a daemon? Discuss the basic coding rules.
33 Explain the terms i)signal ii)signal mask
34 What are daemon processes? Enlist their characteristics. Also write a program to transform a normal user process into a daemon process. Explain every step in the program.
35 Explain the sigaction() function by giving the prototype and discuss its features.
36 Briefly explain the kill() API and alarm() API.
37 List the timer manipulation APIs in POSIX.1b.
38 What is error logging? With a neat block schematic discuss the error login facility in BSD.
39 Briefly explain SIGCHLD Signal and the waitpid API.
40 Discuss daemon characteristics.
41 What are pipes? What are their limitations? Write a c program that sends "hello world" message to the child process through the pipe. The child on receiving this message should

display it on the standard output.

42 With a neat block schematic, explain how FIFO can be used to implement client-server communication model.

43 What are the three different ways in which the client and server processes are can get access to same IPC structures? List the APIs with their argument details that are used to create, control, send and receive messages from a message queue.

44 What are semaphores? What is their purpose?

45 List and explain the APIs used to create and control the semaphores.

46 What are the different system calls available to create and manipulate semaphores? Explain.

47 Explain the concept of shared memory with an example.

48 Explain passing file descriptors over STRAMS-based pipes.

49 What is a socket? Discuss how to create and destroy a socket?

50 Discuss the different functions available for transmitting and receiving data over a socket.

## Lesson Plan

### Unit I

1 **Introduction:** UNIX and ANSI Standards: The ANSI C Standard,
2 The ANSI/ISO C++ Standards, Difference between ANSI C and C++
3 The POSIX Standards, The POSIX.1 FIPS Standard
4 UNIX and POSIX APIs: The POSIX APIs
5 The UNIX and POSIX Development Environment,API Common Characteristics
6 **Unix Files:** File Types, The UNIX and POSIX File System
7 The UNIX and POSIX File Attributes, Inodes in UNIX System V
8 Application Program Interface to Files, UNIX Kernel Support for Files
9 Relationship of C Stream, Pointers and File Descriptors
10 Directory Files, Hard and Symbolic Links.

### Unit II

1 **Unix File Apis:** General File APIs
2 File APIs
3 File and Record Locking
4 Directory File APIs, Device File APIs
5 FIFO File APIs, Symbolic Link File APIs
6 **Unix Processes:** The Environment of a UNIX Process: Introduction, main function
7 Process Termination, Command-Line Arguments, Environment List
8 Memory Layout of a C Program, Shared Libraries, Memory Allocation
9 Environment Variables, setjmp and longjmp Functions
10 getrlimit, setrlimit Functions.

### Unit III

1 **Process Control:** Introduction, Process Identifiers, fork, vfork, exit
2 wait, waitpid, waited, wait3, wait4 Functions,
3 Race Conditions, exec Functions, Changing User IDs and Group IDs,
4 Interpreter Files, system Function, Process Accounting,
5 User Identification, Process Times**,** PROCESS RELATIONSHIPS: Introduction
6 Terminal Logins, Network Logins
7 Process Groups, Sessions
8 Controlling Terminal, tcgetpgrp
9 tcsetpgrp, and tcgetsid Functions, Job Control
10 Shell Execution of Programs, Orphaned Process Groups

### Unit IV

1 **Signals And Daemon Processes:** Signals Introduction

2 The UNIX Kernel Support for Signals,signal
3 Signal Mask, sigaction
4 The SIGCHLD Signal and the waitpid Function
5 The sigsetjmp and siglongjmp Functions, Kill, Alarm
6 Interval Timers, POSIX.lb Timers
7 **Daemon Processes:** Introduction, Daemon Characteristics
8 Coding Rules, Error Logging,
9 Single instance daemons; Daemon conventions;
10 Client-Server Model.

**Unit V**

1 **Interprocess Communication:** Introduction
2 Pipes, examples
3 popen, pclose Functions, examples
4 Coprocesses example
5 FIFOs, examples
6 System V IPC, Message Queues
7 Semaphores, Shared Memory
8 **Network IPC:** Sockets Introduction;
9 Socket Descriptors; Addressing;
10 Connection establishment; Data transfer;
11 Socket options; Out-of-band data;
12 Nonblocking and asynchronous I/O.

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcome (CO)** | | **Program Outcome (ABET/NBA-(3a-m))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Distinguish between ANSI C, K&R C and POSIX Standards | **L3** | | | | | | | | | M | | | | L |
| Describe different UNIX Files | **L2** | | | | | | | | | M | | | | |
| Describe different UNIX APIs | **L2** | | | | | | | | | M | | | | |
| Apply different UNIX APIs depending on the requirement | **L3** | L | | M | M | L | L | | | M | | M | | |
| Implement modules of real world application using appropriate UNIX APIs | **L4** | M | | M | M | M | M | | | M | | H | M | L |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |

| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcome (CO)** | | **Program Outcome (ABET/NBA-(3a-m))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Distinguish between ANSI C, K&R C and POSIX Standards | **L3** | | | | | | | | | 2 | | | | 1 |
| Describe different UNIX Files | **L2** | | | | | | | | | 2 | | | | |
| Describe different UNIX APIs | **L2** | | | | | | | | | 2 | | | | |
| Apply different UNIX APIs depending on the requirement | **L3** | 1 | | 2 | 2 | 1 | 1 | | | 2 | | 2 | | |
| Implement modules of real world application using appropriate UNIX APIs | **L4** | 2 | | 2 | 2 | 2 | 2 | | | 2 | | 3 | 2 | 1 |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |

| Course Title: Data Base Management System | | | |
|---|---|---|---|
| **Course Code: P13IS52** | **Semester : V** | **L- T – P-H : 4 – 0 – 0-4** | **Credits:4** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites:** Operating System and Computer Organization

## Course learning objectives(CLOs)

**This course aims to**

1. State the importance of DBMS and explain how DBMS is better than traditional File Processing Systems.
2. Analyze the basic structure of Database and recognize the different views of the database.
3. Draw and Investigate Data Flow and Entity Relationship Diagrams.
4. Analyze and use Relational Data Model, while comparing with other data models.
5. Formulate data retrieval queries in SQL and the Relational Algebra and Calculus.
6. Understand and explain the terms like Transaction Processing and Concurrency Control.
7. Understand types of database failure and recovery

## Course Content

### Unit-I

**Introduction**

An example: Characteristics of Database approach; Advantages of using DBMS approach; A brief history of database applications; Data models, schemas and instances; Three-schema architecture and data independence; Database languages and interfaces; The database system environment.

**Entity-relationship model:**

Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; Refining the ER Design; ER Diagrams, Naming Conventions and Design Issues; Relationship types of degree higher than two. **10 Hrs**

### Unit-II

**Relational model and relational algebra and basic SQL:**

Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions and dealing with constraint violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations : JOIN and DIVISION; Additional Relational Operations; Examples of Queries in Relational Algebra; Relational Database Design Using ER- to-Relational Mapping.

**SQL- 1:**

SQL Data Definition and Data Types; Specifying basic constraints in SQL; Basic Retrieval Queries in SQL, INSERT, DELETE, and UPDATE Statements in SQL. **12 Hrs**

### Unit-III

**Sql -2: Complex Queries, Triggers, Views, And Schema Modification**

More complex SQL Retrieval Queries, Specifying constraints as Assertion and Actions as Trigger; Views (Virtual Tables) in SQL; Additional features of SQL; Schema Change Statements in SQL; Additional Features of SQL. **10 Hrs**

### Unit-IV

**Database design:**

Informal Design Guidelines for Relation Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form; Multi valued Dependencies and Fourth Normal Form; Join Dependencies and Fifth Normal Form. **10 Hrs**

### Unit – V

**Transaction processing concepts:**

Introduction to Transaction processing; Transactions and System concepts; Desirable properties of transactions; Characterizing Schedules based on Serializability.

**Concurrency control and recovery techniques:**
Two-phase locking techniques for concurrency control; concurrency control based on timestamp ordering; recovery concepts; recovery techniques based on deferred update; recovery techniques based on immediate update; shadow paging. **10 Hrs**

**Text Books:**
1. Fundamentals of Database Systems – Elmasri and Navathe, 6th Edition, Addison-Wesley, 2011

**Reference Books:**
1. Data Base System Concepts – Silberschatz, Korth and Sudharshan, 5th Edition, Mc-GrawHill, 2006
2. An Introduction to Database Systems – C.J. Date, A. Kannan, S.Swamynatham, 8th Edition, Pearson Education, 2006.
3. Database Management Systems – Raghu Ramakrishnan and Johannes Gehrke – 3rd Edition, McGraw-Hill, 2003.

## Course outcomes

**At the end of the course the student should be able to**
1. Explain the fundamental concepts, structures, applications and functions of different components of Databases. Develop the ability to identify required components like entities, attributes and relations for a given mini world description.
2. Design an E-R diagram for the real life application and Construct relational database schema, given the ER diagram of a database application.
3. Construct relational algebraic expressions and Develop SQL to retrieve the database contents.
4. Understand the design process and Apply the suitable normal form to normalize the given data base.
5. Describe the roles of transaction processing in DBMS and the role of concurrency control and recovery techniques in data base design.

## Topic learning objectives

**After learning all the topics of unit-I, the student is able to**
1. Define database and its characteristics -L1
2. Distinguish between conventional file system and Database systems -L2
3. List out the advantages of database approach over conventional file system -L1
4. List out the data base applications .-L1
5. Differentiate between the data models, schemas and instances .-L4
6. Explain the 3-schema architecture of database.-L2
7. Describe the different database languages and interfaces. -L1
8. Describe the different database system components. -L1
9. Identify Entity types, Entity sets, Attributes and Keys from a given application. -L1
10. Design the database using the High-Level Conceptual data models. -L5
11. Differentiate between relationship instance, relationship type and a relationship set.-L4

**After learning all the topics of unit-II, the student is able to**
1. Define Domains, Tuples, Relations, super key, candidate key, and primary key. -L1
2. Explain the relational model concepts and constraints. -L2
3. List the different Database update anomalies.-L1
4. Identify the type of constraints violated by each update operation. -L1
5. Apply the different unary relational algebra operations on Relations.-L3
6. Describe the different set operations using the relational algebra.-L1
7. Differentiate between different types of JOIN operations.-L4
8. Explain the DIVISION operations on Relations.-L2
9. Write a given query in relational algebra.-L5

10. Apply the ER-to-Relational Mapping algorithm to create the Relational Database Schema.-L3
11. Definition of SQL, data definition and data types allowed in SQL.-L1
12. Explain the main types of constraints with examples. -L2

**After learning all the topics of unit-III, the student is able to**

1. Writing the basic and more complex queries in SQL for the given statement.-L5
2. Use of Insert, Delete and Update commands in SQL Statements. -L3
3. List the statements for modifying schemas, Tables and constraint.-L1
4. Explain the Assertion and Trigger; Views (Virtual Tables) in SQL.-L2

**After learning all the topics of unit-IV, the student is able to**

1. Explain informal guidelines for the database design. -L2
2. Illustration with an example for insertion, deletion and modification anomalies. -L4
3. Explain the concept of functional dependency. -L2
4. Apply inference rules for functional dependency. -L3
5. Define 1NF, 2NF, 3NF and BCNF. -L1
6. Design a Normalized database up to BCNF database. -L5
7. Apply normal forms to normalize the given relation schemas.-L3
8. Define Multi valued dependencies 4NF and 5NF. -L1
9. Explain join dependencies, inclusion dependencies and other dependencies.-L2
10. Design a Normalized database up to 5NF for sample example.-L5

**After learning all the topics of unit-V, the student is able to**

1. Explain the concepts of Transaction processing. -L2
2. List desirable properties of Transaction. -L1
3. Describe Schedules and their characteristics.-L1
4. Describe Transaction support in SQL. -L1
5. Explain the necessity of concurrency control. -L2
6. Describe locking and recovery techniques. -L1
7. Explain recovery techniques based on deferred update and immediate update. -L2
8. Differentiate between deferred update and immediate update recovery techniques. -L4
9. Explain the Aries Algorithm. -L2
10. Design schedules based on recoverability and serializability. -L5

**Review questions**

1. Define database and its characteristics
2. Distinguish between conventional file system and Database systems
3. List out the advantages of database approach over conventional file system
4. List out the data base applications .
5. Differentiate between the data models, schemas and instances .
6. Explain the 3-schema architecture of database.
7. Describe the different database languages and interfaces.
8. Describe the different database system components.
9. Identify Entity types, Entity sets, Attributes and Keys from a given application.
10. Design the database using the High-Level Conceptual data models.
11. Differentiate between relationship instance, relationship type and a relationship set.
12. Identify relationships among different entities.
13. Identify Roles, structural constraints and weak entity types.
14. Design an E-R diagram for the given mini world description.
15. Define Domains, Tuples, Relations, super key, candidate key, and primary key.

16. Explain the relational model concepts and constraints.
17. List the different Database update anomalies.
18. Identify the type of constraints violated by each update operation.
19. Apply the different unary relational algebra operations on Relations.
20. Describe the different set operations using the relational algebra.
21. Differentiate between different types of JOIN operations.
22. Explain the DIVISION operations on Relations.
23. Write a given query in relational algebra.
24. Apply the ER-to-Relational Mapping algorithm to create the Relational Database Schema.
25. Definition of SQL, data definition and data types allowed in SQL.
26. Explain the main types of constraints with examples.
27. Construct tables using different data types using SQL command.
28. Writing the basic and more complex queries in SQL for the given statement.
29. Use of Insert, Delete and Update commands in SQL Statements.
30. List the statements for modifying schemas, Tables and constraint.
31. Explain the Assertion and Trigger; Views (Virtual Tables) in SQL.
32. Explain informal guidelines for the database design.
33. Illustration with an example for insertion, deletion and modification anomalies.
34. Explain the concept of functional dependency.
35. Apply inference rules for functional dependency.
36. Define 1NF, 2NF, 3NF and BCNF.
37. Design a Normalized database up to BCNF database.
38. Apply normal forms to normalize the given relation schemas.
39. Define Multi valued dependencies 4NF and 5NF.
40. Design a Normalized database up to 5NF for sample example.
41. Explain the concepts of Transaction processing.
42. List desirable properties of Transaction.
43. Describe Schedules and their characteristics.
44. Describe Transaction support in SQL.
45. Explain the necessity of concurrency control.
46. Describe locking and recovery techniques.
47. Explain recovery techniques based on deferred update and immediate update.
48. Differentiate between deferred update and immediate update recovery techniques.
49. Explain the Aries Algorithm
50. Design schedules based on recoverability and serializability.
51. Interpret a set of transactions to create serializability.

## Lesson plan

### Unit-I

1. Introduction
2. Characteristics of Database approach advantages of using DBMS
3. Data models, schemas and instances.
4. Three-schema architecture and data independence.
5. The database system environment.
6. Entity Types, Entity Sets, Attributes and Keys.
7. Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types.
8. ER Diagrams.
9. Naming Conventions and Design Issues.
10. Relationship types of degree higher than two .

**Unit-II**

1. Relational Model Concepts
2. Relational Model Constraints and Relational Database Schemas.
3. Unary Relational Operations: SELECT and PROJECT
4. Relational Algebra Operations from Set Theory
5. Binary Relational Operations: JOIN
6. Division operator
7. Examples of Queries in Relational Algebra
8. ER- to-Relational Mapping
9. SQL Data Definition and Data Types
10. Basic Retrieval Queries in SQL
11. INSERT, DELETE, and UPDATE Statements in SQL.
12. Examples

**Unit-III**

1. The complex SQL Retrieval Queries
2. Nested queries -examples
3. Correlated queries-examples
4. Correlated queries-examples
5. Assertion
6. Trigger
7. Views (Virtual Tables) in SQL
8. Additional features of SQL
9. Schema Change Statements in SQL
10. Additional Features of SQL

**Unit-IV**

1. Informal Design Guidelines for Relation Schemas
2. Functional Dependencies , Normalization
3. Normal Forms Based on Primary Keys
4. First normal form, second normal form
5. Third normal form
6. Boyce-Codd Normal Form
7. Multi valued Dependencies
8. Fourth Normal Form
9. Join Dependencies
10. Fifth Normal Form.

**Unit-V**

1 Introduction to Transaction processing
2. Transactions and System concepts
3. Desirable properties of transactions
4. Characterizing Schedules based on Serializability
5. Two-phase locking techniques for concurrency control
6. Concurrency control based on timestamp ordering;
7. Recovery techniques based on deferred update
8. Recovery techniques based on immediate update
9. Shadow paging;
10. Exercises.

| **Course Articulation Matrix** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course outcomes** | **Program Outcome (ABET/NBA-(3a-m))** | | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Explain** the fundamental concepts, structures, operations, importance, role and applications and functions of different components of Databases. **Develop** the ability to identify required components like entities, attributes and relations for a given mini world description | **L2 L5** | M | H | L | - | L | - | M | - | L | L | - | - | H |
| **Design** an E-R diagram for the real life application and **Construct** relational database schema, given the ER diagram of a database application. | **L5 L3** | M | H | M | - | H | - | - | L | M | - | M | M | H |
| **Construct** relational algebraic expressions and **Develop** SQL to retrieve the database contents. | **L3** | H | H | M | - | M | - | - | - | - | - | - | L | M |
| **Understand** the design process and **Apply** the suitable normal form to normalize the given data base. | **L3** | M | H | H | - | M | - | - | M | - | - | L | L | - |
| **Describe** the roles of transaction processing in DBMS and the role of concurrency control and recovery techniques in data base design. | **L6** | L | L | M | L | M | - | - | - | - | - | - | - | - |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |

| **Course Assessment Matrix** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course outcomes** | **Program Outcome (ABET/NBA-(3a-m))** | | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Explain** the fundamental concepts, structures, operations, importance, role and applications and functions of different components of Databases. **Develop** the ability to identify required components like entities, attributes and relations for a given mini world description | **L2 L5** | 2 | 3 | 1 | - | 1 | - | 2 | - | 1 | 1 | - | - | - |
| **Design** an E-R diagram for the real life application and **Construct** relational database schema, given the ER diagram of a database application. | **L5 L3** | 2 | 3 | 2 | - | 3 | - | - | 1 | 2 | - | 2 | 2 | 3 |
| **Construct** relational algebraic expressions and **Develop** SQL to retrieve the database contents. | **L3** | 3 | 3 | 2 | - | 2 | - | - | - | - | - | - | 1 | 2 |
| **Understand** the design process and **Apply** the suitable normal form to normalize the given data base. | **L3** | 2 | 3 | 3 | - | 2 | - | - | 2 | - | - | 1 | 1 | - |
| **Describe** the roles of transaction processing in DBMS and the role of concurrency control and recovery techniques in data base design. | **L6** | 1 | 1 | 2 | 1 | 2 | - | - | - | - | - | - | - | - |
| **1 – Low, 2 – Moderate and 3 – High** | | | | | | | | | | | | | | |

| Course Title: Communication Networks | | | |
|---|---|---|---|
| **Course Code: P13IS53** | **Semester : V** | **L- T – P-H : 4 – 0 – 0- 4** | **Credits: 4** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites:** Operating System and Computer Organization

## Course learning objectives(CLOs)

**This course aims to**

1. Understand the Data communication system, Networks and Internet.
2. Understand and recognize the OSI model and TCP/IP model, and also explain the layers in each model.
3. Identify the periodic analog signals and nonperiodic digital signals.
4. Discuss the digital-to-digital conversion, digital-to-analog conversion, analog-to-analog conversion and analog-to-digital conversion techniques.
5. Identify the different types of guided media and unguided media.
6. Identify the different error detection and correction techniques handled by Data link layer.
7. Discuss the taxonomy of protocols.
8. Compare High-level Data Link Control protocol and Point-to-Point protocol.
9. Identify the different multiple access techniques and also discuss the internet.
10. Discuss the wireless LANs, connecting LANs, backbone networks and virtual LANs.

## Course Content

### Unit-I

**Overview Of Networks And Physical Layer**

**Introduction:** Data Communications; Networks; The Internet; Layered tasks; The OSI Model and the layers in the OSI model; TCP / IP Protocol Suite; Addressing; **Physical Layer:** Analog and digital signals; periodic analog signals - sine wave, phase, wavelength, time and frequency domain, bandwidth; Digital signals; Transmission impairment; Data rate limits; Performance. **11 Hrs**

### Unit -II

**Physical Layer Continued**

**Digital Transmission and Analog Transmission:** Digital-to-Digital conversion; Analog-to-Digital conversion; Transmission modes; Digital-to-Analog conversion; Analog-to-Analog conversion; **Transmission media:** Guided Media: Twisted pair cable, Coaxial cable, Fiber-Optic cable; Unguided media: Radio waves, Microwaves, Infrared waves**.** **11 Hrs**

### Unit -III

**Data Link Layer – 1**

**Error Detection and Correction:** Introduction; Block coding; Linear block codes; Cyclic codes - CRC, polynomials; Checksum; **Data Link Control:** Framing; Flow and Error control; Protocols; Noiseless channels; Noisy channels. **10 Hrs**

### Unit -IV

**Data Link Layer – 2**

**Data Link Control:** HDLC; Point-to-point Protocol - framing, transition phases, Multiplexing, Multilink PPP; **Multiple Access:** Random Access; Controlled Access; Channelization. **Wired LANs:** Ethernet: IEEE standards; Standard Ethernet; Fast Ethernet, Gigabit Ethernet. **10 Hrs**

### Unit –V

**Data Link Layer – 3**

**Wireless LANs:** IEE 802.11: Architecture, MAC Sub layer, Addressing Mechanism; Bluetooth: Architecture, Bluetooth layer, Radio layer, Base band layer; **Connecting LANs:** Connecting devices, Backbone Networks, Virtual LANs. **10 Hrs**

**Text Book :**

1. Behrouz A. Forouzan: Data Communications and Networking, 5th Edition, Tata McGraw-Hill, 2013.

**Reference Books :**

1. Alberto Leon-Garcia and IndraWidjaja: Communication Networks - Fundamental Concepts and Key architectures, 3rd Edition, Tata McGraw- Hill, 2006.
2. William Stallings: Data and Computer Communication, 8th Edition, Pearson Education, 2007.

## Course outcomes

1. Describe the OSI model and TCP/IP model, and also explain the layers in each model.
2. Apply digital transmission techniques to transmit the data digitally.
3. Determine the better error detection and correction technique.
4. Analyze the data link control and media access control related to data link layer.
5. Identify the different wireless LANs and connecting devices.

## Topic learning objectives

**After learning all the topics of unit-I, the student is able to**

1. Understand the Data communication system. -L2
2. Identify the different network topologies. -L4
3. Distinguish between LAN and WAN. -L4
4. Describe the history of the Internet and hierarchical organization of the internet. -L1
5. Explain the OSI model. -L2
6. Distinguish between data link layer, transport layer and network layer. -L4
7. Discuss the TCP/IP model. -L6
8. Compare the OSI model and TCP/IP model. -L2
9. Explain the physical address, logical address and port address. -L2
10. Classify the Nyquist bit rate and Shannon capacity according to type of the channel. -L2

**After learning all the topics of unit-II, the student is able to**

1. Describe the three digital-to-digital techniques such as line coding, block coding and scrambling. -L1
2. Discuss the digital-to-analog conversion techniques. -L6
3. Explain the analog-to-analog conversion techniques. -L2
4. Compare PCM and Delta modulation. -L2
5. Distinguish between parallel transmission and serial transmission. -L4
6. Construct the constellation diagram for ASK, BPSK and QPSK. -L5
7. Design the BPSK, BASK and BFSK. -L5
8. Identify the different types of guided media and unguided media. -L4
9. Compare the twisted-pair, coaxial cable and fiber-optic cable. -L2
10. Analyze the electromagnetic spectrum for wireless communication. -L4

**After learning all the topics of unit-III, the student is able to**

1. Explain the types of errors. -L2
2. Identify different error detection and correction techniques handled by Data link layer. -L4
3. Design the structure of block coding. -L5
4. Design the structure of cyclic code. -L5
5. Change a given number to one's complement. -L3
6. Explain the types of framing. -L2
7. Design the protocol with no flow or error control. -L5
8. Write the algorithm for stop-and-wait ARQ. -L5
9. Design the Go-Back-N ARQ. -L5
10. Describe the piggybacking in Go-Back-N ARQ. -L1

**After learning all the topics of unit-IV, the student is able to**

1. Explain the frame formats for HDLC. -L2
2. Describe the transition phases of PPP. -L1
3. Explain the authentication protocols. -L2
4. Compare CSMA/CD and CSMA/CA. -L2

5. Discuss the different channelization protocols. -L6
6. Explain the IEEE standard for LAN. -L2
7. Summarize on the Ethernet evolution. -L5
8. Describe the implementation of the fast Ethernet. -L1
9. Demonstrate the implementation of the gigabit Ethernet. -L3
10. Explain the standard 802.3ae. -L2

**After learning all the topics of unit-V, the student is able to**

1. Construct the flow chart for CSMA/CA. -L5
2. Explain the frame format of MAC layer. -L2
3. Compare hidden and exposed station problems. -L2
4. Distinguish between piconets and scatternet. -L4
5. Describe the layers in the Bluetooth. -L1
6. List the connecting devices. -L1
7. Define the Transparent bridges. -L1
8. Demonstrate the bus backbone and star backbone networks. -L3
9. Explain the virtual LANs. -L2
10. List the advantages of VLANs. -L1

**Review questions**

1. Compare the telephone network and the Internet. What are the similarities? What are the differences?
2. How are OSI and ISO related to each other?
3. Match the following to one or more layers of the OSI model:
   a. Route determination
   b. Flow control
   c. Interface to transmission media
   d. Provides access for the end user
4. What is the difference between network layer delivery and transport layer delivery?
5. What is a peer-to-peer process?
6. What is the difference between a port address, a logical address, and a physical address?
7. How do the layers of the Internet model correlate to the layers of the OSI model?
8. What is the theoretical capacity of a channel in each of the following cases:
   a. Bandwidth: 20 KHz SNRdB=40
   b. Bandwidth: 200 KHz SNRdB=4
   c. Bandwidth: 1MHz SNRdB=20
9. Name three types of transmission impairment.
10. A periodic composite signal contains frequencies from 10 to 30 KHz, each with an amplitude of10 V. Draw the frequency spectrum.
11. List three techniques of digital-to-digital conversion.
12. Distinguish between a signal element and a data element.
13. Distinguish between data rate and signal rate.
14. Draw the graph of the NRZ-L scheme using each of the following data streams, assuming that the last signa11evel has been positive. From the graphs, guess the bandwidth for this scheme using the average number of changes in the signal level.
    a. 00000000
    b. 11111111
    c. 01010101
    d. 00110011
15. What is the maximum data rate of a channel with a bandwidth of200 KHz if we Use four levels of digital signaling.

16. We have a baseband channel with a I-MHz bandwidth. What is the data rate for this channel if we use one of the following line coding schemes?
    a. NRZ-L
    b. Manchester
    c. MLT-3
    d. 2B1Q
17. Distinguish between multilevel TDM, multiple slot TDM, and pulse-stuffed TDM.
18. Distinguish between synchronous and statistical TDM.
19. How do guided media differ from unguided media?
20. What is the difference between omni directional waves and unidirectional waves?
21. What is the definition of a linear block code? What is the definition of a cyclic code?
22. What is the Hamming distance? What is the minimum Hamming distance?
23. How is the simple parity check related to the two-dimensional parity check?
24. What is the Hamming distance for each of the following codewords:
    a. d (10000, 00000)
    b. d (10101, 10000)
    c. d (11111,11111)
    d. d (000, 000)
25. Find the minimum Hamming distance for the following cases:
    a. Detection of two en-ors.
    b. Correction of two errors.
    c. Detection of 3 errors or correction of 2 errors.
    d. Detection of 6 errors or correction of 2 errors.
26. Define framing and the reason for its need.
27. Compare and contrast byte-stuffing and bit-stuffing. Which technique is used in byte-oriented protocols? Which technique is used in bit-oriented protocols?
28. Compare and contrast the Go-Back-NARQ Protocol with Selective-Repeat ARQ.
29. Define piggybacking and its usefulness.
30. Using 5-bit sequence numbers, what is the maximum size of the send and receive windows for each of the following protocols?
    a. Stop-and-WaitARQ
    b. Go-Back-NARQ
    c. Selective-RepeatARQ
31. Compare and contrast HDLC with PPP. Which one is byte-oriented; which one is bit-oriented?
32. What is the difference between a unicast, multicast, and broadcast address?
33. What are the advantages of dividing an Ethernet LAN with a bridge?
34. What is the hexadecimal equivalent of the following Ethernet address?
    01011010 00010001 01010101 00011000 10101010 00001111
35. How does the Ethernet address lA:2B:3CAD:5E:6F appear on the line in binary?
36. How is the preamble field different from the SFD field?
37. 2. What is the purpose of an NIC?
38. What are the common Fast Ethernet implementations?
39. What are the common Gigabit Ethernet implementations?
40. 11. What are the common Ten-Gigabit Ethernet implementations?
41. What is the difference between a BSS and an ESS?
42. What are the two types of links between a Bluetooth primary and a Bluetooth secondary?
43. Compare and contrast CSMA/CD with CSMA/CA.
44. Compare a piconet and a scatternet.
45. What is the purpose of the NAV?
46. What is a transparent bridge?

47. 4. How does a repeater extend the length of a LAN?
48. 5. How is a hub related to a repeater?
49. Create a system of three LANs with four bridges. The bridges (B1to B4) connect the LANs as follows:
    a. B1connectsLAN 1andLAN 2.
    b. B2connectsLAN1andLAN3.
    c. B3 connectsLAN2and LAN 3.
    d. B4connects LAN 1, LAN2, andLAN3.

    Choose B1 as the root bridge. Show the forwarding and blocking ports, after applying the spanning tree procedure.
50. What is the difference between a forwarding port and a blocking port?

## Lesson plan

### Unit -I

1. Introduction: Data Communications; Networks
2. The Internet; Layered tasks
3. The OSI Model
4. Layers in the OSI model
5. TCP / IP Protocol Suite
6. Addressing
7. Physical Layer: Analog and digital signals; periodic analog signals - sine wave, phase
8. Wavelength, time and frequency domain, bandwidth
9. Digital signals; Transmission impairment
10. Data rate limits
11. Performance.

### Unit -II

1. **Digital Transmission and Analog Transmission:** Digital-to-Digital conversion
2. Digital-to-Digital conversion – continued
3. Analog-to-Digital conversion
4. Analog-to-Digital conversion – continued
5. Transmission modes
6. Digital-to-Analog conversion
7. Analog-to-Analog conversion
8. **Transmission media:** Guided Media: Twisted pair cable, Coaxial cable
9. Fiber-Optic cable
10. Unguided media: Radio waves
11. Microwaves, Infrared waves.

### Unit -III

1. **Error Detection and Correction:** Introduction
2. Block coding; Linear block codes
3. Linear block codes – continued
4. Cyclic codes - CRC, polynomials
5. Checksum;
6. **Data Link Control:** Framing; Flow and Error control; Protocols
7. Noiseless channels
8. Noiseless channels – continued
9. Noisy channels
10. Noisy channels – continued.

### Unit -IV

1. **Data Link Control:** HDLC
2. Point-to-point Protocol - framing, transition phases

3. Multiplexing, Multilink PPP
4. **Multiple Access:** Random Access
5. Controlled Access
6. Channelization
7. **Wired LANs:** Ethernet: IEEE standards
8. Standard Ethernet
9. Fast Ethernet
10. Gigabit Ethernet.

**Unit-V**

1. **Wireless LANs:** IEE 802.11: Architecture
2. MAC Sub layer, Addressing Mechanism
3. Bluetooth: Architecture
4. Bluetooth layer, Radio layer
5. Base band layer
6. **Connecting LANs:** Connecting devices
7. Connecting devices – continued
8. Backbone Networks
9. Backbone Networks – continued
10. Virtual LANs.

**Course Articulation Matrix (CAM)**

| Course Outcomes | Program Outcome (ABET/NBA-(3a-k)) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Describe** the OSI model and TCP/IP model, and also **explain** the layers in each model. | **L1 L2 L5** | M | M | M | | L | | | M | | | H | | |
| **Apply** digital transmission techniques to transmit the data digitally. | **L3** | M | M | L | | H | | | L | | | H | L | |
| **Determine** the better error detection and correction technique. | **L3** | H | M | | | H | | | | | L | | | |
| **Analyze** the data link control and media access control related to data link layer. | **L4** | H | H | M | | M | | | | L | | | | |
| **Identify** the different wireless LANs and connecting devices. | **L4** | H | M | M | | L | | | | L | | | | |

**L- Low, M- Moderate, H-High**

**Course Assessment Matrix (CAM)**

| Course Outcomes | Program Outcome (ABET/NBA-(3a-k)) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Describe** the OSI model and TCP/IP model, and also **explain** the layers in each model. | **L1 L2 L5** | 2 | 2 | 2 | | 1 | | | 2 | | | 3 | | |
| **Apply** digital transmission techniques to transmit the data digitally. | **L3** | 2 | 2 | 1 | | 3 | | | 1 | | | 3 | 1 | |
| **Determine** the better error detection and correction technique. | **L3** | 3 | 2 | | | 3 | | | | | 1 | | | |
| **Analyze** the data link control and media access control related to data link layer. | **L4** | 3 | 3 | 2 | | 2 | | | | 1 | | | | |
| **Identify** the different wireless LANs and connecting devices. | **L4** | 3 | 2 | 2 | | 1 | | | | 1 | | | | |

**1 – Low, 2 – Moderate and 3 – High**

| Course Title : System software | | | |
|---|---|---|---|
| **Course Code : P13IS54** | **Semester : V** | **L - T – P-H : 4 - 0 – 0-4** | **Credits: 4** |
| **Contact Period: Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites :**

- Assembly language programming
- Data Structure and algorithms
- Basics of computer organization

## Course Learning Objectives (CLOs)

**This course aims to**

1. Understand the basic machine architecture
2. Design various system software's
3. Know the working principle of assemblers, linkers and loaders.
4. Analyze the working principle of macro processor
5. Understand the application of lex and yacc tools for developing lexical analyzers and parsers.

## Course Content

### Unit - I

**Machine Architecture :**

Introduction to System Software and Machine Architecture, Simplified Instructional Computer (SIC) - SIC Machine Architecture, SIC/XE Machine Architecture, SIC Programming Examples.

CISC machines, RISC machines, comparison of CISC and RISC **9 Hrs**

### Unit-II

**Assemblers :**

Basic Assembler Function - A Simple SIC Assembler, Assembler Algorithm and Data Structures, Machine Dependent Assembler Features - Instruction Formats & Addressing Modes, Program Relocation.

Machine Independent Assembler Features - Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking, Assembler Design Options - One- Pass assembler, Multi-Pass Assembler. **12 Hrs**

### Unit-III

**Loaders and linkers :**

Basic Loader Functions - Design of an Absolute - Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader , Machine-Independent Loader Features - Automatic Library Search, Loader Options, Loader Design Options - Linkage Editor, Dynamic Linking. **10 Hrs**

### Unit-IV

**Macroprocessor-:**

Basic Macro Processor Functions - Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Machine- Independent Macro Processor Features - Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Keyword Macro Parameters Macro processor Design Options – Recursive Macro Expansion, General-Purpose Macro Processors, Macro Processing Within Language Translators, Implementation Examples - MASM Macro Processor, ANSI C Macro Processor. **11Hrs**

### Unit-V

**Lex :**

Recognizing Words With LEX, Symbol Tables, Grammars, Parser-Lexer Communication, The Parts of Speech Lexer, A YACC Parser, The Rules Section, Running LEX and YACC, LEX and Hand- Written Lexers, Examples of Regular Expressions, A Word Counting Program, Parsing a Command Line.

**Yacc :** Using YACC - Grammars, Recursive Rules, Shift/Reduce Parsing, What YACC Cannot Parse; A YACC Parser - The Definition Section, The Rules Section, Symbol Values and Actions, The LEXER, Compiling and Running a Simple Parser, Arithmetic Expressions and Ambiguity, Variables and Typed Token **10 Hrs**

**Text Books :**

1. System Software - an introduction to system programming Leland. L. Beck,D.Manjula 3rd Edition, reprinted in 2013.
2. John. R. Levine, Mason and Doug Brown:Lex and Yacc, O'Reilly, SPD, Reprint March 2005. chapters 1,2(page 27-42),3(page 51-65)

**Reference Book:**

1. System Programming and Operating Systems, D M Dhamdhere, TATA McGraw Hill, 2nd Edition.

## Course Outcomes

**On successful completion of the course the students will be able to**

1. Implement the application using SIC or SIC/XE machine.
2. Generate the machine code for any set of instruction.
3. Analyze the working principles of linkers and loaders.
4. Analyze the working principles of Macro processor.
5. Describe the tools used during lexical and syntax analysis phase of compiler.

## Topic Learning Objectives

**After learning all the topics of unit – I, the student is able to**

1. Define and explain the components of system software. -L1,L2
2. Distinguish between system software and application software. -L2
3. Compare between SIC machine architecture & SIC/XE machine architecture -L4
4. Illustrate how input and output operations are performed in SIC and SIC/XE. -L4
5. Describe how to perform looping and indexing operations of SIC/XE. -L1
6. Explain the architectures of various CISC & RISC machines -L2
7. Differentiate between RISC & CISC machines. -L2
8. Determine the target address for the instruction given. -L3
9. Describe how to perform data movement operations of SIC and SIC/XE machine. -L1
10. Write an SIC or SIC/XE program based on the application given. -L5

**After learning all the topics of unit – II, the student is able to**

1. List basic functions of assemblers. -L1
2. Discuss the data structures used by assemblers. -L2
3. Define machine dependent assembler features. -L1
4. List the advantages and disadvantages of assembly language. -L1,L2
5. Explain the structure of various records used in object program. -L2
6. Apply various assembler directives in programs. -L3
7. Compare two pass assembler with single pass assemblers. -L4
8. Describe the various machine independent assembler features. -L1
9. Justify when we use one pass assembler and why. -L6
10. Compare between program blocks and control sections. -L4

**After learning all the topics of unit – III, the student is able to**

1. Explain the concept of linkers and loaders.-L2
2. List and explain the task of a linker.-L1,L2

3. Explain how relocation is done using bit masking and modification record.-L2
4. Reproduce an algorithm for boot strap loader.- L1
5. Explain how loading and calling a subroutine can be done using dynamic linking.-L2
6. Explain the procedure of performing program linking when sub programs use external references. -L2
7. Justify how ESTAB data structure is used in 2 pass linking loader.-L6
8. Reproduce an algorithm for absolute loader.-L1
9. Determine the object code after performing linking operation.-L4
10. Distinguish between linking loader and linkage editor.- L2

**After learning all the topics of unit – IV, the student is able to**

1. Explain the data structures used in macro processor.-L2
2. Explain machine independent macro processor features.-L2
3. Discuss the problems associated in implementing macros calls within macros.-L6
4. Explain the pros and cons of general purpose macro processor.-L2
5. Analyze various design options of macro processor. -L4
6. Justify why one pass macro processor preferred instead of two pass macroprocessor.-L6
7. Explain the basic functions of macro processor. -L2
8. Decide how macro processor and micro processor varies. -L6
9. Analyze the various procedures used in one pass macro processor. -L4
10. Justify how recursive macro expansion is resolved in one pass macro processor.-L6

**After learning all the topics of unit – V, the student is able to**

1. Identify fundamentals of using LEX.-L1
2. Explain the special characters to describe regular expression in LEX.-L2
3. Describe special purpose routines used by LEX generated state machines.-L1
4. Create a simple application using LEX such as word count in a file, computing statistics on C programs.-L5
5. Conclude how LEX is used to create Compilers and Interpreters.-L6
6. Determine what yacc cannot parse.-L4
7. Explanations about how to build shift reduce parser.-L2
8. Apply yacc in applications -L3
9. Create a simple application using YACC to count no-of if statements in nested-if loop .-L5
10. Difference between lex and yacc structure -L2

## Review Questions

1. Differentiate between system software and application software.
2. Define system software. list few examples of system software
3. Explain the architecture of SIC machine.
4. Explain the architecture of SIC/XE machine.
5. Determine the TA for SIC or SIC/XE machine instruction.
6. Write a SIC/XE program to move a string “INFORMATION “ from one location to another location.
7. Differentiate between RISC or CISC machines.
8. Explain any one architecture of CISC machines.
9. Explain any one architecture of RISC machines.
10. Define why SIC/XE machine is said to be upward compatible.
11. Explain any ten assembler directives with example.
12. Differentiate between literals and constants.
13. What is the information passed from pass1 to pass 2 and why?

14. Write the functions of symbol table and opcode table w.r.t. pass1 and pass2 algorithms.
15. Explain the data structures and variables used in 2pass assembler.
16. Differentiate between program blocks and control section.
17. Explain one pass assembler with examples.
18. Write the difference between absolute expression and relative expression.
19. Write the format for object program.
20. Explain multi pass assembler.
21. Discuss simple bootstrap loader with algorithm.
22. List and explain the different function utilities of loader.
23. Discuss automatic library search with reference to linker and loader.
24. Differentiate between linking loader and linkage editor.
25. Explain pass1 & pass2 algorithms of linking loader.
26. Explain the data structures and variables used in linking loader.
27. Explain how the programs are been linked with examples.
28. Explain absolute loader.
29. Explain dynamic linking with example.
30. List the advantages of dynamic linking.
31. Define macro processor.
32. Explain the basic functions of macro processor.
33. Explain the data structures used in macro processor.
34. List and explain the disadvantage of 2 pass macro processo.
35. Explain the various procedures used in one pass macro processor.
36. Explain any two machine independent macro processor features.
37. Explain recursive macro expansion.
38. Explain the procedures of DEFINE and EXPAND in one pass macro processor
39. List and explain the various macro processor design options.
40. Explain the advantages and disadvantages of general purpose macro processor
41. Explain the structure of a lex program.
42. Define regular expression. Explain all the characters used in regular expression.
43. Define shift reduce parsing? Explain the parsing of the input "fred=12+13" represent it using Parse tree.
44. Write a yacc program to evaluate the expression. Consider all possible cases.
45. Write a lex program to count number of vowels and consonants in a given string
46. Explain parser lexer communication.
47. Explain the structure of the yacc program.
48. Write a LEX specification to count the number of positive, negative integer and positive, negative real number in a given input.
49. Compare and contrast LEX and YACC tools with examples.
50. Define : (i) LEXER (ii) Parser
    (iii) Token (iv) yylex()
51. With reference to LEX and YACC tools.

## Lesson Plan

### Unit – I

1. Introduction to System Software and Machine Architecture
2. Simplified Instructional Computer (SIC) Machine Architecture,
3. SIC/XE Machine Architecture
4. SIC Programming Examples.
5. SIC/XE programming examples
6. CISC machines architecture

7. continuation of CISC machine architecture
8. RISC machines architecture
9. comparison of CISC and RISC and comparison between SIC and SIC/XE machine

**Unit – II**

1. Basic Assembler Function
2. A Simple SIC Assembler, Data Structures
3. Pass1 algorithm
4. Pass2 algorithm
5. Machine Dependent Assembler Features - Instruction Formats & Addressing Modes,
6. Program Relocation.
7. Machine Independent Assembler Features - Literals,Symbol-Definition Statements
8. Various types of expressions
9. Program blocks with examples
10. Control section with examples
11. One pass assembler
12. Multi pass assemblers

**Unit – III**

1 . Basic Loader Functions
2. Design of an Absolute
3. Bootstrap algorithm with explanation
4. Program Relocation with example
5. Program Linking
6. Data Structures for a Linking Loader with pass1 algorithm
7. Pass2 algorithm
8. Machine-Independent Loader Features - Automatic Library Search, Loader Options
9. Loader Design Options – linking loader and Linkage Editor
10. Dynamic Linking with example

**Unit – IV**

1. Basic Macro Processor Functions - Macro Definitions and Expansion
2. Data structures and variables used in one pass algorithm
3. Macro Processor Algorithm
4. Machine- Independent Macro Processor Features - Concatenation of Macro Parameters, Generation of Unique Labels
5. Conditional Macro Expansion
6. Keyword Macro Parameters
7. Macro processor Design Options – Recursive Macro Expansion
8. General-Purpose Macro Processors with its advantages and disadvantages
9. Macro Processing Within Language Translators
10. Implementation Examples - MASM Macro Processor
11. Implementation Examples continuation - ANSI C Macro Processor.

**Unit – V**

1. **LEX :**Recognizing Words With LEX
2. Symbol Tables, Grammars, Parser-Lexer Communication
3. The Parts of Speech Lexer, A YACC Parser, The Rules Section
4. Running LEX and YACC, LEX and Hand- Written Lexers
5. Examples of Regular Expressions, A Word Counting Program, Parsing a Command Line.

6. **YACC :** Using YACC - Grammars, Recursive Rules
7. Shift/Reduce Parsing, What YACC Cannot Parse
8. A YACC Parser - The Definition Section, The Rules Section, Symbol Values and Actions
9. The LEXER, Compiling and Running a Simple Parser
10. Arithmetic Expressions and Ambiguity, Variables and Typed Token

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome(ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Implement the application using SIC or SIC/XE machine | **L1 L6** | L | H | H | | M | | | | H | | | | M |
| Generate the machine code for any set of instruction | **L5 L6** | L | H | H | H | H | | | | H | | M | M | H |
| Analyze the working principles of linkers and loaders | **L4** | L | M | L | | | | | | M | | | M | L |
| Analyze the working principles of Macro processor | **L4** | L | M | L | | | | | | M | | | M | L |
| Describe the tools used during lexical and syntax analysis phase of compiler | **L2 L3** | | H | H | | M | | | | M | | H | | |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcome** | | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Implement the application using SIC or SIC/XE machine | **L1 L6** | 1 | 3 | 3 | | 2 | | | | 3 | | | | 2 |
| Generate the machine code for any set of instruction | **L5 L6** | 1 | 3 | 3 | 3 | 3 | | | | 3 | | 2 | 2 | 3 |
| Analyze the working principles of linkers and loaders | **L4** | 1 | 2 | 1 | | | | | | 2 | | | 2 | 1 |
| Analyze the working principles of Macro processor | **L4** | 1 | 2 | 1 | | | | | | 2 | | | 2 | 1 |
| Describe the tools used during lexical and syntax analysis phase of compiler | **L2 L3** | | 3 | 3 | | 2 | | | | 2 | | 3 | | |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |

| Course Title: Software Engineering | | | |
|---|---|---|---|
| **Course Code: P13IS55** | **Semester : V** | **L- T – P-H : 2 – 2 – 0-4** | **Credits:3** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50% ; SEE: 50%** | |

## Course learning objectives

**This course aims to**

1. Study a body of knowledge relating to Software Engineering, Software reengineering, and maintenance;
2. Understand the principles of large scale software systems, and the processes that are used to build them;
3. Use tools and techniques for producing application software solutions from informal and semi-formal problem specifications;
4. Acquire and develop many valuable skills such as the ability to use computer aided software
5. Evaluate requirements for a software system
6. Apply the process of analysis and design using object oriented approach.
7. Communicate to others the progress of the system development and the contents of the design by means of reports and presentations.
8. Recognize current trends in the area of software engineering
9. Identify the processes, techniques and deliverables that are associated with requirement engineering including system requirement and system modeling
10. Identify the importance of testing in assuring the quality of software with an understanding of managing risks during the progress of the project.

## Course contents

### Unit-I

**Overview, and Requirements**

Introduction: FAQ's about software engineering, Professional and ethical responsibility; software process models, process iteration, software specification, software design and implementation, software validation, software evaluation; Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; the software requirements document; requirements engineering processes: feasibility studies, requirements elicitation and analysis, requirement validation and management; system models: context models, behavioral model, data models, object models, CASE workbenches; software prototyping: prototyping in the software process, rapid prototyping techniques, user interface prototyping. **12 Hrs**

### Unit-II

**Software Design**

Architectural Design: system structuring, control models, modular decomposition, domain-specific architectures; object oriented design: Objects and Object Classes, An Object-Oriented design process; Design evolution; user interface design: user interface design principles, user interactions, information presentation, user support, interface evolution. **10 Hrs**

### Unit-III

**Critical System, Verification and Validation**

Dependability: critical systems, availability and reliability, safety, security; critical system specification: software reliability specification, safety specification, security specification; critical system development: fault tolerance, fault tolerant architectures safe system design; verification and validation: Verification and Validation: Planning; Software inspections; Automated static analysis, clean room software development; software testing: defect testing, integration testing, object oriented testing, testing workbenches. **10Hrs**

### Unit-IV

**Management**

Managing People: limits to thinking, group working, choosing and keeping people, the people capability maturity model; software cost estimation: productivity, estimation techniques, algorithmic cost

modeling, project duration and staffing; quality management: quality assurance and standards, quality planning, quality control, software measurement and metrics; process improvement: process and product quality, process analysis and modeling, process measurement, the SEI process capability maturity model.

**10 Hrs**

**Unit- V**

**Evolution**

Legacy Systems: legacy system structures, legacy system design, legacy system assessment; software change: program evolution dynamics, software maintenance, architectural evolution; software Re-engineering: source code translation, reverse engineering, program structure improvement, program modularization, data re-engineering **10 Hrs**

**Text book:**

1. Software Engineering – Ian Somerville, 8th Edition, Pearson Education, 2007.

**Reference books :**

1. Software Engineering: A Practitioners Approach **-** Roger S. Pressman, 7th Edition,McGraw-Hill, 2007.
2. Software Engineering Theory and Practice - Shari Lawrence Pfleeger, Joanne M. Atlee, 3rd Edition, Pearson Education, 2006.
3. Software Engineering Principles and Practice – Waman S Jawadekar, Tata McGraw Hill, 2004
4. Software Engineering – Pankaj Jalote, Tata Mc Graw Hill.

**Course outcomes**

1. **Demonstrate** an understanding of the principles and techniques of Software Engineering
2. **Analyze** the various steps involved in the design process and the different design approaches which include function-oriented design and object-oriented design
3. **Understand** the activities in project management, requirement engineering process and to identify the different types of system models
4. **Apply** the knowledge of design engineering in software development
5. **Provide** an understanding of the principles of software engineering in a broader system context and the notions of software engineering process and management.

**Topic learning objectives**

**After learning all the topics of unit-I, the student is able to**

1. Understand what software engineering is and why it is important;L2
2. Know the answers to key questions which provide an introduction to software engineering;L1
3. Understand ethical and professional issues which are important for software engineer.L2
4. Understand the concept of a software process and a software process model.L2
5. Understand the number of different software process models and when might be used.L2
6. Understand the concepts of user requirements and system requirements L2
7. Describe the user requirement and system requirement notation.L1
8. Discuss the principle requirement engineering activities and their relationshipsL6
9. Derive the importance of requirement validation.L6

**After learning all the topics of unit-II, the student is able to**

1. Understand why the architectural design of software is important;L2
2. Introduce a number of different types of software architecture covering system structure, and modular decompositionL4
3. Discuss the models required to document system architecture.L6
4. Know the most important activities in a general object oriented design processL1
5. Introduce to the representation of models in unified modeling language.L4

6. Understand different models that may be used to document an object oriented designL2
7. Be aware of five different styles of interaction with a software systemL1
8. Understand usability attributes and simple approaches to system evolution.L2
9. Introduce to different presentation styles of information presentationL4

**After learning all the topics of unit-III, the student is able to**

1. Introduce the notation of critical system L4
2. Understand four dimensions of dependabilityL2
3. Introduce number of metrics for reliabilityL4
4. Understand how safety requirement for critical system may be derived from an analysis of hazards and risks.L2
5. Understand the software development that help avoid introducing faults into a software system.L2
6. Introduce N-versions programming and recovery blocks.L4
7. Distinguish between software verification and software validation.L2
8. Describe the program inspection as a method of discovering defects in programs.L1
9. Understand the clean room method of program development and why it can be effective.L2

**After learning all the topics of unit-IV, the student is able to**

1. Discuss the issues of team workingL6
2. Describe issues involved in selecting and retaining staff in a software developmentL1
3. Understand the fundamentals of software costing and pricing and the complex relationship between them;L2
4. Introduce three metrics that are used for software productivity assessment.L4
5. Appreciate that a range of different techniques should be used when estimating software costs and schedule.L2
6. Understand the principle of COCOMO 2 model for algorithmic cost estimation.L2
7. Understand the principles of software process improvement and why process improvement is worthwhile;L2
8. Discuss the CMM modelL6

**After learning all the topics of unit-V, the student is able to**

1. Understand what is meant by a legacy system and know why these systems are critical to the operation of many businesses.L6
2. Introduce common legacy system structures.L4
3. Understand how legacy system can be assessed to decide if they should be scrapped maintained re-engineered.L2
4. Understand the principles of software maintenance and why software is to expensive to maintainL2
5. Understand why re-engineering is sometimes a cost effective options for software system evolution.L2
6. Understand the activities such as reverse engineering and program restructuring L2
7. Differentiate software and data re-engineeringL4
8. Understand why data re-engineering is an expensive and time consuming processL2
9. Understand hoe legacy system may be transformed to distributed client-server system to extend their life L2

**Review Questions**

1. What is software engineering?
2. What is the difference between program and software?
3. Write out the reasons for the Failure of Water Fall Model.
4. What are the characteristics of the software?
5. Define the terms : Agility and Agile Team

6. What are the challenges in software?
7. Define software process
8. What are the fundamental activities of a software process?
9. What are the umbrella activities of a software process?
10. What are the merits of incremental model?
11. List the task regions in the Spiral model.
12. What are the drawbacks of spiral model?
13. List the process maturity levels in SEIs CMM.
14. What is an effectors process?
15. Define the computer based system.
16. What does Verification represent?
17. What does Validation represent?
18. What is the difference between the "Known Risks" and Predictable Risks"?
19. Explain about the incremental model.
20. Explain in detail about the life cycle process.
21. Explain Spiral model and win-win spiral model in detail?
22. Define software prototyping.
23. What are the benefits of prototyping?
24. What are the advantages of evolutionary prototyping?
25. What are the various Rapid prototyping techniques?
26. What is System Modeling?
27. What are the characteristics of SRS?
28. What is ERD? What is DFD?
29. What does Level0 DFD represent?
30. What is a state transition diagram?
31. Explain in detail about Functional Modeling.
32. Explain in detail about Structural Modeling.
33. Explain about rapid prototyping techniques.
34. Explain the prototyping approaches in software process.
35. How the Architecture Design can be represented?
36. Define design process. List the principles of a software design.
37. What are the common activities in design process?
38. What are the various elements of data design?
39. Define White Box Testing?
40. What is Regression Testing?
41. What are the reasons behind to perform white box testing?
42. What are the approaches of integration testing?
43. Explain in detail about Black box testing.
44. Explain about the software testing strategies.
45. Write short note on the various estimation techniques.
46. What is COCOMO model?
47. What is the purpose of timeline chart?
48. Define CASE Tools.
49. What is software maintenance?
50. What is DFD? State the notations used for creating a DFD.
51. Differentiate between Software Verification & Software Validation.
52. Explain top-down approach used in Integration Testing.

## Lesson Plan

### Unit-I

1. Introduction: FAQ's about software engineering, Professional and ethical responsibility software process models.
2. Process iteration, software specification, software design and implementation.
3. Software validation, software evaluation; Software Requirements.
4. Functional and Non-functional requirements; User requirements.
5. System requirements; the software requirements document
6. Requirements engineering processes: feasibility studies, requirements elicitation and analysis.
7. Requirement validation and management; system models.
8. Context models, behavioural model, data models, object models
9. CASE workbenches.
10. Software prototyping: prototyping in the software process
11. Rapid prototyping techniques
12. User interface prototyping.

### Unit-II

1. Architectural Design: system structuring.
2. Control models, modular decomposition.
3. Domain-specific architectures
4. Object oriented design: Objects and Object Classes
5. An Object-Oriented design process
6. Design evolution
7. User interface design: user interface design principles
8. User interactions
9. Information presentation
10. User support, interface evolution

### Unit-III

1. Dependability: critical systems
2. Availability and reliability
3. Safety, security.
4. Critical system specification: software reliability specification
5. Safety specification, security specification
6. Critical system development: fault tolerance, fault tolerant architectures safe system design
7. Verification and validation: Verification and Validation: Planning; Software inspections; Automated static analysis
8. Clean room software development
9. Software testing: defect testing, integration testing
10. Object oriented testing, testing workbenches

### Unit- IV

1. Managing People: limits to thinking.
2. Group working, choosing and keeping people.
3. The people capability maturity model; software cost estimation: productivity.
4. Estimation techniques, algorithmic cost modeling, project duration and staffing.
5. Quality management: quality assurance and standards.
6. Quality planning, quality control, software measurement and metrics.
7. Process improvement: process and product quality.
8. Process analysis and modelling.
9. Process measurement.
10. The SEI process capability maturity model.

**Unit- V**

1. Legacy Systems: legacy system structures
2. Legacy system design
3. Legacy system assessment
4. Software change: program evolution dynamics
5. Software maintenance
6. Architectural evolution
7. Software Re-engineering: source code translation
8. Reverse engineering
9. Program structure improvement
10. Program modularization, data re-engineering

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Demonstrate** an understanding of the principles and techniques of Software Engineering | **L1** | M | L | M | - | M | - | - | - | - | M | M | - | H |
| **Analyze** the various steps involved in the design process and the different design approaches which include function-oriented design and object-oriented design | **L6** | M | - | - | M | M | - | - | - | - | H | M | M | - |
| **Understand** the activities in project management,requirement engineering process and to identify the different types of system models | L2 | M | M | - | - | - | - | - | - | - | M | M | - | M |
| **Apply** the knowledge of design engineering in software development | **L3** | H | H | H | M | - | - | - | - | - | H | M | M | - |
| **Generalize** an understanding of the principles of software engineering in a broader system context and the notions of software engineering process and management | **L2** | M | M | M | - | M | - | - | - | - | M | M | - | - |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcomes** | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Demonstrate** an understanding of the principles and techniques of Software Engineering | **L1** | 2 | 1 | 2 | - | 2 | - | - | - | - | 2 | 2 | - | 3 |
| **Analyze** the various steps involved in the design process and the different design approaches which include function-oriented design and object-oriented design | **L6** | 2 | - | - | 2 | 2 | - | - | - | - | 3 | 2 | 2 | - |
| **Understand** the activities in project management, requirement engineering process and to identify the different types of system models | **L2** | 2 | 2 | - | - | - | - | - | - | - | 2 | 2 | - | 2 |
| **Apply** the knowledge of design engineering in software development | **L3** | 3 | 3 | 3 | 2 | - | - | - | - | - | 3 | 2 | 2 | - |
| **Provide** an understanding of the principles of software engineering in a broader system context and the notions of software engineering process and management | **L2** | 2 | 2 | 2 | - | 2 | - | - | - | - | 2 | 2 | - | - |
| **1 – Low, 2 – Moderate and 3 – High** | | | | | | | | | | | | | | |

| Course Title: Management and Entrepreneurship | | | |
|---|---|---|---|
| **Course Code: P13IS56** | **Semester : V** | **L- T – P-H : 4 – 0 – 0- 4** | **Credits: 4** |
| **Contact period : 52 Hrs, Exam: 3 Hrs** | **Weightage: CIE: 50%; SEE: 50%** | | |

## Course learning objectives(CLOs)

**This course aims to**

1. Understand the roles and functions of management.
2. Analyze the evolution of management from classical approach to modern approach.
3. Differentiate between strategic planning and tactical planning.
4. Explain the recruitment and selection process.
5. Compare formal and informal communication.
6. Identify the new control techniques.
7. Explain the role of entrepreneurs in the economic development of the nation.
8. Identify the contents of the project report.
9. Understand the different Industry Policy Resolutions.
10. Understand the rules, policies and working criteria by visiting small scale industry.

## Course Content

### Unit-I

**Management**

**Nature and Functions of Management :** Importance and definition of management, management functions, roles of a manager, levels of management, managerial skills, management and administration , management – a science or art or profession, professional management versus family management. **Development of Management Thought:** Early classical approaches, Neo classical approaches, modern approaches. **10 Hrs**

### Unit-II

**Planning:** Importance of planning, forms and types of plans, steps in planning, limitations of planning, making planning effective, planning skills. Decision making: Types of decisions, steps in Rational decision making, rationality in decision making, environment of decision making. **Organisation:** Process of organizing, span of management, principles of organising, Departmentalisation, committees, teams. Staffing: Manpower planning, recruitment and selection process. **11 Hrs**

### Unit-III

**Direction:** Requirements of effective direction, giving orders, motivation. Communication: Purposes of communication, formal communication, forms of communication, informal communication, barriers to communication. Leadership: characteristics of leadership, functions of leader.

**Control**: Steps in control process, need for control, types of control methods, essentials of effective control system, new control techniques. **10 Hrs**

### Unit-IV

**Entrepreneurship: Introduction,** Entrepreneur, Functions of an Entrepreneur, Types of Entrepreneur, Intrapreneur, Entrepreneurship, Evolution of Entrepreneurship, Development of Entrepreneurship, Stages in entrepreneurial process, Role of entrepreneurs in Economic Development, Entrepreneurship in India, Entrepreneurship – its Barriers, women entrepreneurs, Entrepreneurship Development Programme. Preparation of project: meaning of project, project identification, project selection, project report need and significance, contents of project report, project formulation**.** **10 Hrs**

### Unit-V

**Small Scale Industry**: Definition, Characteristics, Need and rationale, Objectives, Scope, role and advantages of SSI, steps to start an SSI, Different Industry Policy Resolutions (IPRs) of S.S.I, Government Support for S.S.I. during 5 year plans, Impact of Liberalization, Privatization, Globalization on S.S.I., WTO/GATT, supporting Agencies of Government for S.S.I, Ancillary Industry and Tiny

Industry. Institutional Support: State level institutions, Central Government institutions. **Case Study:** Students in batch has to visit the nearest SSI and finally submit a report. **11 Hrs**

**Text Book :**

1 Principles of Management, P C Tripathi, P N Reddy, 5th edition, TataMcGraw Hill, 2012.
2 Management and Entrepreneurship, N V R Naidu, T Krishna Rao, I K International Publishing House Pvt. Ltd., 2010.

**Reference Books :**

1. Dynamics of Entrepreneurial Development & Management - Vasant Desai, Himalaya Publishing House, 2007.
2. Management Fundamental Concepts, Application, Skill Development - Robert Lusier - Thomson, 2007.

**Course outcomes**

**After learning all the units, the student is able to**

1. Understand the evolution of management from classical approach to modern approach.
2. Explain the steps in planning and steps in selection process.
3. Identify the direction and control techniques.
4. Compare Entrepreneur and Intrapreneur.
5. Analyze the government support for small scale industry.

**Topic learning objectives**

**After learning all the topics of unit-I, the student is able to**

1. Define management. –L1
2. Describe the roles of a manager. –L6
3. List the various levels of management. –L1
4. Describe the advantages of human relations movement. –L6
5. Explain the principles of administrative management. –L2
6. Discuss the important roles of manager. –L6
7. Compare Taylor's approach and Fayol's approach. –L4
8. Distinguish between Family management and Professional management. –L4
9. Explain the Systems approach. –L2
10. Describe management is a science or an art. –L6

**After learning all the topics of unit-II, the student is able to**

1. Give suggestions to make committees effective. –L2
2. Differentiate between strategic planning and tactical planning. –L4
3. Explain the principles of organizing. –L2
4. Define span of management. –L1
5. Explain the importance of planning. –L2
6. Distinguish between groups and teams. –L4
7. List the types of plans. –L1
8. Explain the steps in planning. –L2
9. Summarize the types of Departmentalisation. –L6
10. Explain the rationality in decision making. –L2

**After learning all the topics of unit-III, the student is able to**

1. Explain the requirements of effective direction. –L2
2. Explain the importance of job satisfaction. –L2
3. Describe the methods of establishing control. –L6
4. Differentiate between direction and supervision. –L4

5. Compare the different motivation theories. –L4
6. Explain the types of formal communication. –L2
7. Compare formal and informal communication. –L4
8. Describe the process of giving orders. –L6
9. Describe the steps in control process. –L6
10. Explain the essentials of sound control system. –L2

**After learning all the topics of unit-IV, the student is able to**

1. Differentiate between Intrapreneur and Entrepreneur. –L4
2. Explain the stages involved in Entrepreneurial process. –L2
3. Categorize the different types of Entrepreneurs. –L5
4. Explain the barriers of entrepreneurship. –L2
5. Explain the role of entrepreneurs in the economic development of a nation. –L2
6. Categorize the different types of Entrepreneurs. –L5
7. Describe the functions of an Entrepreneur. –L6
8. Explain entrepreneurship in India. –L2
9. Explain the barriers of entrepreneurship. –L2
10. List the functions of women entrepreneurs. –L1

**After learning all the topics of unit-V, the student is able to**

1. Define SSI, Ancillary industry and Tiny industry. –L1
2. Explain need and rationale of SSIs. –L2
3. Describe the scope of SSIs. –L6
4. Explain the impact of liberalization, privatization and globalization on SSI. –L2
5. Describe GATT. –L6
6. Explain WTO and its impact on trade. –L2
7. Discuss the necessity of institutional support to the SSIs. –L6
8. Explain the role of DICs. –L2
9. Compare ICICI and SIDBI. –L4
10. Explain the need for support to SSIs. –L2

**Review questions**

1. Define management.
2. Explain the importance of management
3. Describe the roles of a manager
4. Differentiate between management and administration
5. List the various levels of management
6. Describe the skills of a manager
7. Explain the principles of administrative management
8. Explain bureaucratic management.
9. Describe the advantages of human relations movement.
10. Explain the systems approach
11. List the types of plans
12. Explain the importance of planning
13. Describe the stages of planning
14. Differentiate between strategic planning and tactical planning
15. Explain the limitations of planning
16. Explain the principles of organizing
17. Describe the various types of departmentalization.
18. Define span of management

19. Differentiate between the different organization structure
20. Give suggestions to make committees effective
21. Explain the requirements of effective direction
22. Compare the different motivation theories
23. Explain the importance of job satisfaction
24. Describe the process of giving orders
25. Differentiate between direction and supervision
26. Explain the importance of organizational commitment
27. Explain the first level supervision
28. Describe the steps in control process
29. Explain the essentials of sound control system
30. Describe the methods of establishing control
31. Define Entrepreneurship
32. Differentiate between Intrapreneur and Entrepreneur
33. Explain the stages involved in Entrepreneurial process.
34. Describe the characteristics of an Entrepreneur.
35. Explain the role of entrepreneurs in the economic development of a nation.
36. Categorize the different types of Entrepreneurs
37. Describe the functions of an Entrepreneur
38. Explain entrepreneurship in India
39. Explain the barriers of entrepreneurship
40. Describe the evolution of entrepreneurship
41. Define SSI, Ancillary industry and Tiny industry.
42. Explain need and rationale of SSIs.
43. Describe the scope of SSIs.
44. Explain the impact of liberalization, privatization and globalization on SSI.
45. Describe GATT.
46. Explain WTO and its impact on trade.
47. Discuss the necessity of institutional support to the SSIs.
48. Explain the role of DICs.
49. Compare ICICI and SIDBI.
50. Explain the need for support to SSIs.

## Lesson plan

### Unit-I

1. Nature and Functions of Management : Importance and definition of management
2. Management functions, roles of a manager
3. Levels of management, managerial skills
4. Management and administration, management – a science or art or profession
5. Professional management versus family management.
6. Development of Management Thought: Early classical approaches
7. Early classical approaches – continued
8. Neo classical approaches
9. Modern approaches
10. Modern approaches – continued

### Unit-II

1. Planning: Importance of planning, forms and types of plans
2. Steps in planning, limitations of planning
3. Making planning effective, planning skills
4. Decision making: Types of decisions, steps in Rational decision making
5. Rationality in decision making

6. Environment of decision making.
7. Organisation: Process of organizing, span of management
8. Principles of organizing
9. Departmentalisation, committees, teams
10. Staffing: Manpower planning
11. Recruitment and selection process.

**Unit-III**

1. Direction: Requirements of effective direction, giving orders
2. Motivation
3. Communication: Purposes of communication, formal communication
4. Forms of communication, informal communication, barriers to communication
5. Leadership: characteristics of leadership, functions of leader.
6. Control: Steps in control process
7. Need for control,
8. Types of control methods
9. Essentials of effective control system
10. New control techniques.

**Unit-IV**

1. Entrepreneurship: Introduction, Entrepreneur
2. Functions of an Entrepreneur, Types of Entrepreneur
3. Intrapreneur, Entrepreneurship, Evolution of Entrepreneurship
4. Development of Entrepreneurship, Stages in entrepreneurial process
5. Role of entrepreneurs in Economic Development, Entrepreneurship in India
6. Entrepreneurship – its Barriers, women entrepreneurs, Entrepreneurship Development Programme.
7. Preparation of project: meaning of project, project identification
8. Project selection, project report need and significance
9. Contents of project report
10. Project formulation.

**Unit-V**

1. Small Scale Industry: Definition, Characteristics, Need and rationale
2. Objectives, Scope, role and advantages of SSI
3. Steps to start an SSI
4. Different Industry Policy Resolutions (IPRs) of S.S.I
5. Government Support for S.S.I. during 5 year plans
6. Impact of Liberalization, Privatization, Globalization on S.S.I., WTO/GATT,
7. Supporting Agencies of Government for S.S.I
8. Ancillary Industry and Tiny Industry
9. Institutional Support: State level institutions
10. Central Government institutions
11. Institutional support continued

| Course Articulation Matrix (CAM) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Understand the evolution of management from classical approach to modern approach. | **L2** | | | | | | H | H | H | L | H | | | |
| Explain the steps in planning and steps in selection process. | **L2** | | L | | | | | M | M | - | H | L | | |
| Identify the direction and control techniques. | **L1** | | L | | | | H | H | H | L | H | | | |
| Compare Entrepreneur and Intrapreneur | **L4** | | | | | | | L | H | L | M | L | | |
| Analyze the government support for small scale industry. | **L4** | | | | | | H | M | L | L | H | | | L |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcomes** | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Understand the evolution of management from classical approach to modern approach. | **L2** | | | | | | 3 | 3 | 3 | 1 | 3 | | | |
| Explain the steps in planning and steps in selection process. | **L2** | | 1 | | | | | 2 | 2 | | 3 | 1 | | |
| Identify the direction and control techniques. | **L1** | | 1 | | | | 3 | 3 | 3 | 1 | 3 | | | |
| Compare Entrepreneur and Intrapreneur. | **L4** | | | | | | | 1 | 3 | 1 | 2 | 1 | | |
| Analyze the government support for small scale industry. | **L4** | | | | | | 3 | 2 | 1 | 1 | 3 | | | 1 |
| **1 – Low, 2 – Moderate and 3 – High** | | | | | | | | | | | | | | |

| **Course Title: System Software and Script lab** | | | |
|---|---|---|---|
| **Course Code: P13ISL57** | **Semester : V** | **L- T – P-H : 0 – 0 – 3-3** | **Credits:1.5** |
| **Contact period : 39 Hrs Exam : 3Hrs** | | **Weightage: : CIE: 50%; SEE: 50%** | |

**Prerequisites:** System Software, LEX and YACC and UNIX Script

**Course learning objectives(CLOs)**

**This course aims to**

1. Understand to generate tokens for given program.
2. Understand to check syntax of the higher level language statements using yacc tools
3. Understand to write a script for any application.

**LEX PROGRAMS**

1. LEX program to display only C comments in a given C file.
2. LEX program to count all instance of She and He including the instances of He that are included in She.
3. LEX program to identify integer numbers, floating point numbers, keywords, identifiers operators in a given input file.
4. LEX program which counts number of words with length one characters, two char so on in a given input file.
5. LEX program to recognize the valid arithmetic expression and identify identifiers and operators present and print them separately.
6. LEX program to recognize and count the number of identifier in a given file.

7. LEX program to extract all HTML tags in a given file.
8. LEX program to count number of characters, words, spaces and lines in a given input file.

**YACC PROGRAMS**

1. YACC program to recognize a valid arithmetic expression that uses the operator +,-,*,/.
2. YACC program to recognize the nested if control statement and display the number of levels of nesting.
3. YACC program to recognize the grammar $A^nB^n$ where n>=0.
4. YACC program to recognize the grammar $A^mB^n$ where n,m>=0.
5. YACC program to recognize the validity of if then statement or if then else statement.
6. YACC program to check the validity of an address (format: first line is name (FNAME, MIDNAME LNAME) and second line is door number and street third line city state pin code.
7. YACC program to evaluate to given arithmetic expression involving operators +,-,* and /.
8. YACC program to recognize variables letters followed by digits, letters should be lower case.

**SHELL PROGRAMS**

1. SHELL script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.
2. SHELL script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. SHELL script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
4. SHELL script that takes a command –line argument and reports on whether it is directory, a file, or something else.
5. SHELL script that computes the gross salary of a employee according to the following rules:
   a. If basic salary is < 1500 then HRA =10% of the basic and DA =90% of the basic.
   b. If basic salary is >=1500 then HRA =Rs500 and DA=98% of the basic salary is entered interactively through the key board.
6. SHELL script that accepts two integers as its arguments and compute the value of first number raised to the power of the second number.
7. SHELL script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.

SHELL script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.

| **Course Title: Data Base Management System Lab** | | | |
|---|---|---|---|
| **Course Code: P13ISL58** | **Semester : V** | **L- T – P -H: 0-0-3-3** | **Credits:1.5** |
| **Contact period** : **39 Hrs Exam : 3Hrs** | | | **Weightage**: **CIE:50%; SEE:50%** |

**Prerequisites:** Know about SQL, ORACLE and VISUAL BASIC languages

**Course learning objectives(CLOs)**

**This course aims to**

1. Provide a strong formal foundation in database concepts, technology and practice to the students.
2. Familiarize the students with the database environments towards an information-oriented data-processing oriented framework.
3. Understand the relational data model and to introduction to systematic database design approaches covering conceptual design, logical design
4. Present the concepts and techniques relating to query processing by SQL engines
5. Develop a database application using any of the commercial application product (DB2, Visual Basic,

**PART-A**

1. **Consider the customer-sale scenario given below. The primary keys are underlined and the data types are specified:**

   CUSTOMER(**Cust id : integer,** cust_name: string)
   ITEM(**item_id: integer,** item_name: string, price: integer)
   SALE(**bill_no: integer,** bill_data: date, **cust_id: integer, item_id: integer**, qty_sold: integer)

   For the above schema, perform the following:
   a) Create the tables with the appropriate integrity constraints
   b) Insert around 10 records in each of the tables
   c) List all the bills for the current date with the customer names and item numbers
   d) List the total Bill details with the quantity sold, price of the item and the final amount
   e) List the details of the customer who have bought a product which has a price>200
   f) Give a count of how many products have been bought by each customer
   g) Give a list of products bought by a customer having cust_id as 5
   h) List the item details which are sold as of today
   i) Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold, amount
   j) Create a view which lists the daily sales date wise for the last one week.

2. **Consider the Student Library scenario given below. The primary keys are underlined and the data types are specified:**

   STUDENT(**Stud_no : integer,** Stud_name: string)
   MEMBERSHIP(**Mem_no: integer**, **Stud_no: integer**)
   BOOK(**book_no: integer**, book_name:string, author: string)
   ISS_REC(**iss_no:integer**, iss_date: date, **Mem_no: integer**, **book_no: integer**)

   For the above schema, perform the following:
   a) Create the tables with the appropriate integrity constraints
   b) Insert around 10 records in each of the tables
   c) List all the student names with their membership numbers
   d) List all the issues for the current date with student and Book names
   e) List the details of students who borrowed book whose author is CJDATE
   f) Give a count of how many books have been bought by each student
   g) Give a list of books taken by student with stud_no as 5
   h) List the book details which are issued as of today
   i) Create a view which lists out the iss_no, iss _date, stud_name, book name
   j) Create a view which lists the daily issues-date wise for the last one week

3. **Consider the Employee-pay scenario given below. The primary keys are underlined and the data types are specified:**

   EMPLOYEE(**emp_id : integer,** emp_name: string)
   DEPARTMENT(**dept_id: integer**, dept_name:string)
   PAYDETAILS(**emp_id : integer**, **dept_id: integer**, basic: integer, deductions: integer, additions: integer, DOJ: date)
   PAYROLL(**emp_id : integer**, pay_date: date)

   For the above schema, perform the following:
   a) Create the tables with the appropriate integrity constraints
   b) Insert around 10 records in each of the tables
   c) List the employee details department wise
   d) List all the employee names who joined after particular date
   e) List the details of employees whose basic salary is between 10,000 and 20,000

f) Give a count of how many employees are working in each department
g) Give a names of the employees whose netsalary>10,000
h) List the details for an employee_id=5
i) Create a view which lists out the emp_name, department, basic, dedeuctions, netsalary
j) Create a view which lists the emp_name and his netsalary

4. **Consider the Insurance database given below. The primary keys are underlined and the data types are specified:**

PERSON (driver – id #: String, name: string, address: string) CAR (regno: string, model: string, year: int)
ACCIDENT (report-number: int, accd-date: date, location: string)
OWNS (driver-id #:string, Regno:string)
PARTICIPATED (driver-id: string, Regno:string, report-number:int, damage amount:int)

a) Create the above tables by properly specifying the primary keys and the foreign keys.
b) Enter at least five tuples for each relation.
c) Demonstrate how you
   i. Update the damage amount to 25000 for the car with a specific regno in the ACCIDENT table with report number 12.
   ii. Add a new accident to the database.
d) Find the total number of people who owned cars that were involved in accidents in 2008
e) Find the number of accidents in which cars belonging to a specific model were involved.
f) Generate suitable reports.
g) Create suitable front end for querying and displaying the results.

5. **Consider the following relations for an order processing database application in a company**

CUSTOMER (cust #: int , cname: string, city: string)
ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)
ORDER – ITEM (order #: int, item #: int, qty: int)
ITEM (item # : int, unit price: int)
SHIPMENT (order #:int,warehouse#: int, ship-date:date)
WAREHOUSE (warehouse #: int, city: string)

a) Create the above tables by properly specifying the primary keys and the foreign keys
b) Enter at least five tuples for each relation.
c) Produce a listing CUSTNAME ,#ofOrders , AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.
d) List the order# for orders that were shipped from all the warehouses that the company has in a specific city.
e) Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER_ITEM table that contain this particular item.
f) Generate suitable reports.
g) Create suitable front end for querying and displaying the results.

6. **Consider the following database of student enrollment in courses & books adopted for each course:**

STUDENT(regno: string, name: string, major: string, bdate:date)
COURSE (course #:int, cname:string, dept:string)
ENROLL ( regno:string, course#:int, sem:int, marks:int)
BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)
TEXT(book-ISBN:int,book-title:string,publisher:string, author:string)

a) Create the above tables by properly specifying the primary keys and the foreign keys.
b) Enter at least five tuples for each relation.
c) Demonstrate how you add a new text book to the database and make this book be adopted by some department.
d) Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
e) List any department that has all its adopted books published by a specific publisher.
f) Generate suitable reports.
g) Create suitable front end for querying and displaying the results.

**7. The following tables are maintained by a book dealer:**

AUTHOR (author-id:int, name:string, city:string, country:string)
PUBLISHER (publisher-id:int, name:string, city:string, country:string)
CATALOG (book-id:int, title:string, author-id:int, publisher-id:int, category-id:int, year:int, price:int)
CATEGORY (category-id:int, description:string)
ORDER-DETAILS (order-no:int, book-id:int, quantity:int)

a) Create the above tables by properly specifying the primary keys and the foreign keys.
b) Enter at least five tuples for each relation.
c) Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
d) Find the author of the book which has maximum sales.
e) Demonstrate how you increase the price of books published by a specific publisher by 10%.
f) Generate suitable reports.
g) Create suitable front end for querying and displaying the results.

**8. Consider the following database for a banking enterprise:**

BRANCH(branch-name:string,branch-city:string,assets:real)
ACCOUNT(accno:int,branch-name:string,balance:real)
DEPOSITOR(customer-name:string, accno:int)
CUSTOMER(customer-name:string, customer-street:string, customer- city:string)
LOAN(loan-number:int, branch-name:string, amount:real) BORROWER(customer-ame:string, loan-number:int)

a) Create the above tables by properly specifying the primary keys and the foreign keys
b) Enter at least five tuples for each relation
c) Find all the customers who have at least two accounts at the Main branch.
d) Find all the customers who have an account at all the branches located in a specific city.
e) Demonstrate how you delete tuples in ACCOUNT relation at every branch located in a specific city.
f) Generate suitable reports.
g) Create suitable front end for querying and displaying the results.

**PART – B**

**Implement the any project compulsory using DBMS Concepts with suitable front end design**

| Course Title: Object Oriented System Development | | | |
|---|---|---|---|
| **Course Code: P13IS61** | **Semester : VI** | **L- T – P -H: 4 – 0 – 0-4** | **Credits:4** |
| **Contact period : 52 Hrs Exam: 3Hrs** | | **Weightage: : CIE:50% ; SEE:50%** | |

**Prerequisites:**

A good understanding of object oriented technologies and a basic understanding of analysis and design.

## Course learning Objectives (CLO's)

**This course aims to**

1. Understand the concepts of OO.
2. Describe Unified Modeling Language (UML) notations.
3. Describe the concepts and notations involved in OO modeling.
4. Explain the models for Object Oriented System Development.
5. Apply UML notations to OO models.
6. Analyze and Design a solution to the real world problem.
7. Understand the basics of how to prepare OO models.
8. Discuss the significance of the design patterns.
9. Use object-oriented software metrics to characterize developed software.
10. Apply software metrics to the improve software design.

## Course Content

### Unit - I

**Introduction, Modeling Concepts, Class Modeling And Advanced Class Modeling:** What is Object Orientation? What is OO development? OO themes; **Modeling as Design Technique**: Modeling; Abstraction; The three models.

**Class Modeling:** Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model; Navigation of class models.

**Advanced Class Modeling:** Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages. **11 Hrs**

### Unit - II

**State Modeling, Advanced State Modeling, And Interaction Modeling**: **State Modeling**: Events, States, Transitions and Conditions; State diagrams; State diagram behavior.

**Advanced State Modeling:** Nested state diagrams; Nested states; Signal generalization; Concurrency; A sample state model; Relation of class and state models;

**Interaction Modeling**: Use case models; Sequence models; Activity models.

**Advanced Interaction Modeling**: Use case relationships; Procedural sequence models; Special constructs for activity models**.** **10 Hrs**

### Unit - III

**System conception, domain analysis, application analysis:**

**System Conception:** Devising a system concept; Elaborating a concept; Preparing a problem statement.

**Domain Analysis:** Overview of analysis; Domain class model; Domain state model; Domain Interaction model; Iterating the analysis

**Application Analysis**: Application interaction model; Application class model; Application state model; Adding operations. **10 Hrs**

### Unit - IV

**System Design, Class Design:**

**System design**: Overview of system design; Estimating performance; Making a reuse plan; Breaking a system in to sub-systems; Identifying concurrency; Allocation of sub-systems; Management of data storage; Handling global resources; Choosing a software control strategy; Handling boundary

conditions; Setting the trade-off priorities; Common architectural styles; Architecture of the ATM system as the example.

**Class Design**: Overview of class design; Bridging the gap; Realizing use cases; Designing algorithms; Recursing downwards, Refactoring; Design optimization; Reification of behavior; Adjustment of inheritance; Organizing a class design; ATM example. **11 Hrs**

**Unit - V**

**Design Patterns:** What is a pattern and what makes a pattern? Pattern categories; Relationships between patterns; Pattern description Communication Patterns: Forwarder-Receiver; Client-Dispatcher-Server; Publisher-Subscriber. Management Patterns: Command processor; View handler.

**Metrics:** Product quality metrics; In-process quality metrics; Metrics for software maintenance; Design and complexity metrics; Productivity metrics; Quality and quality management metrics; Lessons learned for OO projects. **10 Hrs**

**Text Books**

1.Object-Oriented Modeling and Design with UML - Michael Blaha, James Rumbaugh , 2nd Edition, Pearson Education, 2012.
2.Pattern-Oriented Software Architecture, A System of Patterns: Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal:, Volume 1, John Wiley and Sons, 2007.
3.Metrics and Models in Software Quality Engineering, Stephan H. Kan:, 2nd Edition, Pearson.

**Reference Books**

1. Object-Oriented Analysis and Design with Applications – Grady Booch et al, 3rd edition, Pearson education, 2007.
2 Object-Oriented Analysis, Design, and Implementation, Brahma Dathan, Sarnath Ramnath, Universities Press, 2009.
3. Software Metrics: A Rigorous Approach, Fenton N. E., S. L. Pfleeger, 2nd Edition, Cengage Learning.

**Course Outcomes**

**After learning all the units of the course, the student is able to**

1. Understand object-oriented modeling concepts and class model.
2. Explain state model and interaction model with UML notations.
3. Analyse to build domain and application model.
4. Design solutions for real world problems.
5. Apply design patterns to solve real world problems and also use metrics to evaluate the designed software.

**Topic learning Objectives(TLO's)**

**After learning all the topics of unit – I, the student is able to**

1. Understand object oriented concepts. -L1
2. Define modeling. -L1
3. Explain three kinds of models-class, state and interaction. -L2
4. Describe the class model, which deals with the structural "data" aspects of a system. -L2
5. Describe class modeling concepts like object, class, link, association generalization and inheritance. -L2
6. Apply UML notations for class model concepts with examples. -L3
7. Explain how to navigate a class model and uncovers hidden flaws and omissions using OCL. -L2
8. Understand few advanced class modeling concepts for improved modeling and for complex applications. -L2
9. Explain aggregation and composition with example. -L2
10. Describe Multiple Inheritances. -L2

**After learning all the topics of unit – II, the student is able to**

1. Describe the state model, which concerns the control aspects of a system. -L2
2. Describe how the state model specifies and implements. -L2
3. Apply UML notations for state model concepts with example. -L3
4. Describe what an object does in response to events. -L2
5. Understand how to model complex systems by using nested state diagrams, nested states, signal generalization and concurrency. -L2
6. Sample state diagram for programmable thermostat. -L3
7. Explain how class and state model are related. -L2
8. Describe how objects interact to produce useful results.-L2
9. Explain the interaction model representation starts with use cases that are then elaborated with sequence and activity diagrams. -L2
10. Describe advanced interaction modeling features that can be helpful in complex applications. -L2

**After learning all the topics of unit – III, the student is able to**

1. Describe the first stage of development-system conception.-L1
2. Understand how to devise a new system. -L1
3. Understand the problem and prepare a problem statement with ATM case study. -L2
4. Understand how to take OO concepts and apply them to construct a domain model. -L2
5. Understand how to build a domain model that focuses on the real world things that carry the semantics and behaviour of the application. -L2
6. Describe how to analyse the problem iteratively. -L2
7. Understand building an application model that addresses the computer aspects of the application that are visible to users. -L2
8. Understand the steps to follow to construct an application interaction model. -L2
9. Understand how to specify user interface, define class boundaries and determine controllers. -L2
10. Understand the steps to follow to construct an application state model. -L2

**After learning all the topics of unit – IV, the student is able to**

1. Describe how to address system design. -L3
2. Explain the task of partitioning a system into subsystems and making high-level policy decisions. -L2
3. Understand many aspects that should be consider when formulating a system design. -L2
4. Explain how to make a reuse plan. -L2
5. Explain two kinds of control flows in a software system. -L2
6. Describe several common architectural styles in existing system. -L2
7. Describe the architecture of the ATM system. -L3
8. Understand how the class design adds details, such as designing algorithms, refactoring operations, optimizing classes, adjusting inheritance and refining packages. -L2
9. Explain two ways of downward recursion. -L2
10. Construct final ATM domain class model. -L4

**After learning all the topics of unit – V, the student is able to**

1. Define Pattern and explain how it is categorized – L1, L2.
2. Explain Communication Patterns with an example – L2.
3. List out and explain Management Patterns – L1, L2.
4. Define Metric and explain how it is categorized – L1, L2.
5. Explain In-process quality metrics with an example – L2.
6. List out and explain metrics for software maintenance – L1, L2.
7. Explain Design and complexity metrics with an example – L2.
8. Explain Productivity metrics with an example – L2.

## Review Questions

1. What is object orientation? Explain its aspects with an example.
2. Explain the three kinds of models in OO development.
3. What is a model? What purpose does it serve
4. Briefly explain the different kinds of models to describe a system?
5. Write a class model of a library management system.
6. Use examples and explain the following with UML notations.
   i. Class ii. Object iii. Association end names iv. Ordering the object for an association end
7. Write a class model of a window system.
8. Explain aggregation and composition with example?
9. Explain multiple inheritances in detail?
10. Define the following terms, with an example
    i..Abstract classes ii. Meta data iii. Derived data
11. Define event in state modelling. Explain the different kinds of events.
12. Write a state diagram for telephone line.
13. Explain aggregate concurrency and concurrency with in an object w.r.t state diagram with an example.
14. Explain do, entry and exit activities in state diagram behaviour.
15. Explain Synchronization of concurrent activities w.r.t state diagram, with an example.
16. In detail, explain activity diagram with UML notation. Give an example.
17. Draw a use case diagram for a vending machine. What the guidelines needed to be followed while drawing use case model.
18. Explain extend and include use case relationship with example.
19. Explain activity diagram with example. Give guidelines for activity models.
20. Explain sequence diagram with passive objects and transient objects.
21. List some ways to find new system concepts.
22. Explain how a problem statement is prepared.
23. Write and explain the steps performed in constructing a domain state model, with an example.
24. With ATM example, describe how to identify classes in constructing a domain class model.
25. Explain the steps to be followed to construct an application interaction model.
26. With suitable example, explain the steps followed to construct an application state model.
27. What is scenario? Give normal ATM scenario for initial session, process transaction and query account.
28. What are the high level questions generally needed to be asked for an ATM system.
29. Draw the activity diagram for card verification for an ATM.
30. Describe the steps to construct an application class model.
31. List the various decisions to be made during system design. Explain any two.
32. Identify the considerations for choosing among alternative algorithm.
33. What are the two different aspects of reuse? Explain reusable things.
34. Explain how the system is break into subsystems.
35. Explain two kinds of control flows in software systems.
36. Explain the different steps to perform for designing algorithm.
37. List and explain two main ways of downward recursion.
38. Explain batch transformation and continuous transformation architectural styles suited for system design.
39. Describe the steps to improve the organization of a class design.
40. List and explain the tasks involved in design optimization.
41. Define Pattern.
42. Explain how patterns are categorized.
43. Explain Communication Patterns with an example.

44. List out and explain Management Patterns.
45. Define Metric.
46. Explain how metrics are categorized.
47. Explain In-process quality metrics with an example.
48. List out and explain metrics for software maintenance.
49. Explain Design and complexity metrics with an example.
50. Explain Productivity metrics with an example.

## Lesson plan

### Unit I

1. What is Object Orientation? What is OO development?
2. OO themes; Modeling; Abstraction; The three models
3. Object and class concepts; Link and associations concepts
4. Generalization and inheritance
5. A sample class model
6. Navigation of class models.
7. Advanced object and class concepts; Association ends
8. N-ary associations; Aggregation
9. Abstract classes; Metadata
10. Multiple inheritance; Reification
11. Constraints; Derived data; Packages

### Unit II

1. Events, States; Transitions and Conditions
2. State diagrams; State diagram behaviour
3. Nested state diagrams; Nested states
4. Signal generalization; Concurrency
5. A sample state model; Relation of class and state models
6. Use case models
7. Sequence models
8. Activity models
9. Use case relationships; Procedural sequence models
10. Special constructs for activity models

### Unit III

1. Devising a system concept; Elaborating a concept
2. Preparing a problem statement.
3. Overview of analysis; Domain class model
4. Domain class model
5. Domain state model
6. Domain interaction model
7. Iterating the analysis
8. Application interaction model
9. Application class model
10. Application state model; Adding operations.

### Unit IV

1. Overview of system design; Estimating performance
2. Making a reuse plan
3. Breaking a system in to sub-systems; Identifying concurrency
4. Allocation of sub-systems; Management of data storage; Handling global resources
5. Choosing a software control strategy; Handling boundary conditions; Setting the trade-off priorities
6. Common architectural styles; Architecture of the ATM system as the example.

7. Overview of class design; Bridging the gap; Realizing use cases
8. Designing algorithms
9. Recursing downwards, Refactoring
10. Design optimization; Reification of behavior
11. Adjustment of inheritance; Organizing a class design; ATM example

**Unit V**

1. What is a pattern and what makes a pattern? Pattern categories
2. Relationships between patterns; Pattern description Communication Patterns
3. Forwarder-Receiver; Client-Dispatcher-Server
4. Publisher-Subscriber. Management Patterns: Command processor
5. View handler, Metrics Introduction
6. Product quality metrics; In-process quality metrics
7. Metrics for software maintenance; Design and complexity metrics
8. Productivity metrics; Quality and quality management metrics
9. Lessons learned for OO projects.
10. Examples

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome(ABET/NBA-(3a-m))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Understand object-oriented modeling concepts and class model. | **L2** | | M | L | | L | | | | M | | M | | L |
| Explain state model and interaction model with UML notations. | **L2** | | M | L | | L | | | | M | | M | | L |
| Analyse to build domain and application model. | **L3** | | M | L | L | M | | | | M | | M | | L |
| Design solutions for real world problems. | **L3** | | M | M | M | M | | | | H | | M | | M |
| Apply design patterns to solve real world problems and also use metrics to evaluate the designed software. | **L3** | M | M | M | M | M | | | | H | | M | M | L |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcomes** | | **Program Outcome(ABET/NBA-(3a-m))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Understand object-oriented modeling concepts and class model. | **L2** | | 2 | 1 | | 1 | | | | 2 | | 2 | | 1 |
| Explain state model and interaction model with UML notations. | **L2** | | 2 | 1 | | 1 | | | | 2 | | 2 | | 1 |
| Analyse to build domain and application model. | **L3** | | 2 | 1 | 1 | 2 | | | | 2 | | 2 | | 1 |
| Design solutions for real world problems. | **L3** | | 2 | 2 | 2 | 2 | | | | 3 | | 2 | | 2 |
| Apply design patterns to solve real world problems and also use metrics to evaluate the designed software. | **L3** | 2 | 2 | 2 | 2 | 2 | | | | 3 | | 2 | 2 | 1 |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |

| Course Title: Computer Networks | | | |
|---|---|---|---|
| **Course Code: P13IS62** | **Semester : VI** | **L- T – P-H : 4 – 0 – 0-4** | **Credits:4** |
| **Contact period : Lecture: 52 Hr, Exam: 3 Hrs** | | **Weightage: CIE: 50% ; SEE: 50%** | |

**Prerequisites:** Data communication

**Course learning objectives(CLOs)**

**This course aims to**

1. Understand fundamental underlying principles of computer networking.
2. Understand details and functionality of layered network architecture.
3. Enumerate the layers of the OSI model and TCP/IP. Explain the functions of each layer.
4. Understand and recognize the importance of application layer and the working of protocols like HTTP, FTP, DNS etc.
5. Differentiate between TCP & UDP, and understand the services provided by them like flow control, congestion control etc.
6. Identify the different types of multimedia and analyze the quality of service provided by them.
7. Identify some of the factors driving the need for network security, and Identify and classify particular examples of attacks.
8. Compare and contrast symmetric and asymmetric encryption systems and their vulnerability to attack.
9. Identify types of firewall implementation suitable for differing security requirements.

**Course Content**

**Unit-I**

**Introduction And Network Layer – 1**

A brief history: Arpanet, birth of the internet ,transmission control protocol/internetworking protocol (TCP/IP) ,MILNET, CSNET, NSFNET, ANSNET the internet today ,world wide web, time line growth of the internet; protocols and standards: protocols, standards; standards organizations: standards creation committees ,forums, regulatory agencies; internet standards : maturity levels ,requirement levels, internet administration; the OSI model and the TCP/IP protocol suite: protocol layers ,the OSI model, TCP/IP protocol suite, addressing; **Network Layer**: introduction to network layer, switching, packet switching at network layer, network layer services, other network layer issues; IPv4 addresses: introduction, classful addressing, classless addressing, special addresses, NAT; delivery and forwarding of IP packets: delivery, forwarding, structure of a router. **9 Hrs**

**Unit-II**

**Network Layer -2**

Internet protocol version 4 (IPv4) : introduction, datagram, fragmentation, options, checksum; address resolution protocol (ARP) :address mapping, the ARP protocol; mobile IP: addressing, agents, three phases, inefficiency in mobile IP; unicast routing protocols: RIP,OSPF, BGP; multicasting and multicast routing protocols: introduction, multicast addresses, IGMP

Introduction, Virtual Circuit and Datagram Networks, Internet Protocol (IP), Routing algorithms, Link state routing algorithms, Distance vector routing algorithms, Hierarchical routing, Routing in the Internet, Broadcast and Multicast Routing. **11 Hrs**

**Unit-III**

**Transport Layer**

Transport-layer services, transport-layer protocols; user datagram protocol (UDP) :introduction, user datagram, UDP services, UDP applications; transmission control protocol (TCP) : TCP services, TCP features, segment, a TCP connection, state transition diagram, windows in TCP, flow control, error control, congestion control, TCP timers; stream control transmission protocol (SCTP): introduction, SCTP services SCTP features, packet format, an SCTP association, state transition diagram, flow control, error control, congestion control. **10 Hrs**

**Unit - IV**

**Application Layer**

Introduction to the application layer :client-server paradigm, peer-to-peer paradigm ; host configuration: DHCP: introduction, DHCP operation, configuration; domain name system (DNS): need for DNS, name space, DNS in the internet, resolution, DNS messages, types of records; remote login: telnet and SSH: telnet, secure shell (SSH); file transfer: ftp and TFTP: ftp, TFTP; world wide web and http :architecture, web documents, http; electronic mail: SMTP, pop, IMAP, and mime: architecture, user agent, message transfer agent: SMTP, message access agent: POP and IMAP, MIME; multimedia: introduction, digitizing audio and video, audio and video compression ,streaming live audio/video, voice over IP, quality of services. **12 Hrs**

**Unit - V**

**Ipv6 Addressing And Security**

IPv6 addressing :introduction, address space allocation, global unicast addresses, auto configuration, renumbering; IPv6 protocol: introduction, packet format, transition from ipv4 to IPv6; cryptography and network security: introduction, traditional ciphers, modern ciphers, asymmetric-key ciphers, message integrity, message authentication, digital signature, entity authentication, key management; internet security: network layer security, transport layer security, application layer security, firewalls. **10 Hrs**

**Text Books:**

1. Behrouz A. Forouzan: "TCP/IP Protocol Suite" Fourth Edition Tata McGraw-Hill, 2010
2. James F. Kurose and Keith W. Ross: Computer Networking: A Top-Down Approach, 5th edition, Addison-Wesley, 2009. (unit-II)

**Reference Books:**

1. Alberto Leon-Garcia and Indra Widjaja: Communication Networks – Fundamental Concepts and Key architectures, 2nd Edition, Tata McGraw-Hill, 2004.
2. Andrew S Tennanbaum: Computer Networks, 4th Edition, PHI Series.
3. Behrouz A. Forouzan: Data Communications and Networking, 4th Edition, Tata McGraw-Hill, 2006.

**Course outcomes**

**After learning all the units of the course, the student is able to**

1. Discuss different internet standards.
2. Compare various routing algorithms like distance vector, link state, hierarchial & multicast routing, and understand the concept of fragmentation. Use different protocols to achieve Address mapping, Error reporting, unicast routing and multicasting
3. Differentiate between TCP & UDP, and understand the services provided by them like flow control, congestion control etc.
4. Understand and recognize the importance of application layer and the working of protocols like HTTP, FTP and DNS. Identify the different types of multimedia and analyze the quality of service provided by them.
5. Identify and solve the problems associated with transition from IPV4 to IPV6. understand the factors driving the need for network security

**Topic learning objectives**

**After learning all the topics of unit-I, the student is able to**

1. Describe a brief history of the Internet. -L1
2. To define the definition of the two often-used terms in the discussion of the Internet: protocol and standard. - L1,L6
3. To define Internet Standards and explain the mechanism through which these standards are developed. - L1

4. Explain the OSI model and its layer architecture and to show the interface between the layers. - L5, l2
5. To analyze the TCP/IP protocol suite and compare its layers with the ones in the OSI model. -L4
6. To show the functionality of each layer in the TCP/IP protocol with some examples.-L2
7. To introduce switching and in particular packet switching as the mechanism of data delivery in the network layer. - L4
8. To discuss two distinct types of services a packet-switch network can provide: connectionless service and connection-oriented service. -L6
9. To evaluates how routers forward packets in a connectionless packet switch network using the destination address of the packet and a routing table. -L6
10. To discuss how routers forward packets in a connection-oriented packet-switch network using the label on the packet and a routing table. -L6
11. To introduce the concept of an address space in general and the address space of IPv4 in particular. -L4
12. To discuss the classful architecture, classes in this model, and the blocks of addresses available in each class.- L6
13. To discuss NAT technology and show how it can be used to improve the shortage in number of addresses in IPv4. -L6 L2

**After learning all the topics of unit-II, the student is able to**

1. To explain the general idea behind the IP protocol and show the position of IP in relation to other protocols in TCP/IP protocol suite.- L6
2. To show the general format of an IPv4 datagram and list the fields in the header.-L2
3. To show how a checksum is calculated for the header of an IPv4 datagram at the sending site and how the checksum is checked at the receiver site. -L2
4. To show how the address resolution protocol (ARP) is used to dynamically map a logical address to a physical address. -L2
5. To discuss how the Routing Information Protocol (RIP) is used to implement the idea of distance vector routing in the Internet. -L6, L4
6. To discuss how Open Shortest Path First (OSPF) is used to implement the idea of link state routing in the Internet. -L6, L4
7. To discuss how Border Gateway Protocol (BGP) is used to implement the idea of path vector routing in the Internet. -L6, L4
8. To discuss and differentiate between virtual circuit and datagram network.-L6
9. Analyze the link state routing algorithms.-L4
10. Discuss and apply distance vector routing and hierarchical algorithm for finding feasible path. -L6,L3
11. Differentiate between broadcast and multicast routing.-L4

**After learning all the topics of unit-III, the student is able to**

1. To define process-to-process communication at the transport layer and compare it with host-to-host communication at the network layer. -L1, L4
2. To discuss the addressing mechanism at the transport layer, to discuss port numbers, and to define the range port numbers used for different purposes.-L6,L1
3. To illustrates UDP services and show its relationship to other protocols in the TCP/IP protocol suite.-L2, L4
4. To explain the format of a UDP packet, which is called a user datagram, and discuss the use of each field in the header. -L6
5. To discuss TCP as a protocol that provides reliable stream delivery service. -L6

6. To define TCP features and compare them with UDP features.-L1, L4
7. To define the format of a TCP segment and its fields.-L1
8. To show how TCP provides a connection-oriented service, and show the segments exchanged during connection establishment and connection termination phases.-L2
9. To discuss the state transition diagram for TCP and discuss some scenarios. -L6
10. To explain the flow control, error control and congestion control mechanism in TCP
11. To discuss SCTP services and compare them with TCP. -L6,L4
12. To compare and contrast the state transition diagram of SCTP with the corresponding diagram of TCP. -L4
13. To explain the flow control, error control and congestion control mechanism in SCTP and discuss the behavior of the sender site and the receiver site. -L6

**After learning all the topics of unit-IV, the student is able to**

1. To define DHCP as the current Dynamic Host Configuration Protocol. -L1
2. To describe the purpose of DNS. -L6
3. To introduce the TELNET protocol and show how it implements local and remote login using the concept of network virtual terminal. -L4 L2
4. To show how different components of SSH are combined to provide a secure connection over an insecure TCP connection. -L2
5. To discuss FTP and two connections used in this protocol: control connection and data connection. -L6
6. To explain three types of file transfer transferred by FTP. -L6
7. To discuss five types of TFTP messages and their applications.
8. To discuss the architecture of WWW and describe the concepts of hypertext and hypermedia. -L6
9. To introduce three different Web documents: static document, dynamic document, and active document. -L5
10. To introduce cookies and their applications in HTTP. -L4
11. To explain the mechanism of sending and receiving e-mails. -L6
12. To demonstrates e-mail transfer phases.- L3
13. To discuss two message access agents (MAAs): POP and IMAP. -L6
14. To show how audio/video files can be downloaded for future use or broadcast to clients over the Internet. The Internet can also be used for live audio/video interaction. Audio and video need to be digitized before being sent over the Internet.-L2
15. To discuss voice over IP as a real-time interactive audio/video application.-L6

**After learning all the topics of unit-V, the student is able to**

1. To use the format of an IPv6 datagram discuss composed of a base header and a payload.-L2,L6
2. To discuss different fields used in an IPv6 datagram based header and compare them with the fields in IPv4 datagram. -L6
3. To introduce security goals and to discuss the types of attacks that threatens these goals.-L4
4. To introduce traditional ciphers as symmetric-key ciphers to create the background for understanding modern symmetric-key ciphers.-L4
5. To introduce the elements of modern block ciphers and show an example of a modern block cipher in which these elements are used.-L4
6. To discuss the general idea behind asymmetric-key ciphers and introduce one common cipher in this category. -L6
7. To show how secret keys in symmetric-key cryptography and how public keys in asymmetric-key cryptography can be distributed and managed using KDCs or certificate authorities (CAs).

-L2

8. To introduce the idea of Internet security at the network layer and the IPSec protocol that implements that idea in two modes: transport and tunnel. -L4
9. To discuss two protocols in IPSec, AH and ESP, and explain the security services each provide. -L6
10. To introduce security association and its implementation in IPSec. -L4

**Review questions**

1. Use the Internet to find the number of RFCs.
2. How are OSI and ISO related to each other?
3. Match the following to one or more layers of the OSI model:

a. route determination
b. flow control
c. interface to transmission media
d. provides access for the end user

4. If a port number is 16 bits (2 bytes), what is the minimum header size at transport layer, network layer of the TCP/IP protocol suite?
5. Do we encapsulate our message when we send a regular letter to a friend? When we send a post card to a friend while we are vacationing in another country, do we encapsulate our message?
6. Why do you think a radio station does not need the addresses of its listeners when a message is broadcast?
7. The data rate of 10Base5 is 10 Mbps. How long does it take to create the smallest frame? Show your calculation.
8. Using the data in Exercises 3 and 4, find the minimum size of an Ethernet frame for collision detection to work properly. An Ethernet MAC sub layer receives 42 bytes of data from the LLC sub layer. How many bytes of padding must be added to the data?
9. An Ethernet MAC sub layer receives 1510 bytes of data from the LLC layer. Can the data be encapsulated in one frame? If not, how many frames need to be sent? What is the size of the data in each frame?
10. Compare and contrast CSMA/CD with CSMA/CA.
11. Host A and host B use IPSec in the transport mode. Can we say that the two hosts need to create a virtual connection-oriented service between them? Explain.
12. When we talk about authentication in PGP (or S/MIME), do we mean message authentication or entity authentication? Explain.
13. If cryptography algorithms in PGP or S/MIME cannot be negotiated, how can the receiver of the e-mail determine which algorithm have been used by the sender?
14. Can we use SSL with UDP? Explain.
15. Why is there no need for security association with SSL?
16. Compare and contrast PGP and S/MIME. What are the advantage and disadvantage of each?
17. If Alice and Bob are continuously sending messages to each other, can they create a security association once and use it for every packet exchanged? Explain.
18. Should the handshaking in SSL occur before or after the three-way handshaking in TCP? Can they be combined? Explain.
19. Define the type of the attack in each of the following cases:

a. A student breaks into a professor's office to obtain a copy of the next test.
b. A student gives a check for $10 to buy a used book. Later the student finds out that the check was cashed for $100.
c. A student sends hundreds of e-mails per day to the school using a phony return e-mail address.

20. Alice can use only the additive cipher on her computer to send a message to a friend. She thinks that the message is more secure if she encrypts the message two times, each time with a

different key. Is she right? Defend your answer.

21. Encrypt the message "this is an exercise" using additive cipher with key = 20. Ignore the space between words. Decrypt the message to get the original plaintext.
22. Find the result of the following operations:
    a. (01001101) ⊕(01001101)
    b. (01001101) ⊕(10110010
23. In RSA, given n = 12091, e = 13, and d = 3653 encrypt the message "THIS IS TOUGH" using the 00 to 26 encoding scheme. Decrypt the ciphertext to find the original message. Use 4-digit plaintext cipher text blocks.
24. In RSA, why can't Bob choose 1 as the public key e?
25. Explain why private-public keys cannot be used in creating a MAC.
26. Assume we have a very simple message digest. Our unrealistic message digest is just one number between 0 and 25. The digest is initially set to 0. The cryptographic hash function adds the current value of the digest to the value of the current character (between 0 and 25). Addition is in modulo 26. What is the value of the digest if the message is "HELLO"? Why is this digest not secure?
27. A sender sends a series of packets to the same destination using 5-bit sequence of numbers. If the sequence number starts with 0, what is the sequence number of the 100th packet?
28. Using 5-bit sequence numbers, what is the maximum size of the send and receive windows for each of the following protocols?
    a. Stop-and-Wait
    b. Go-Back-N
    c. Selective-Repeat
29. In cases where reliability is not of primary importance, UDP would make a good transport protocol. Give examples of specific cases.
30. Are both UDP and IP unreliable to the same degree? Why or why not?
31. A TFTP server residing on a host with IP address 130.45.12.7 sends a message to a TFTP client residing on a host with IP address 14.90.90.33. What is the pair of sockets used in this communication?
32. Compare the TCP header and the UDP header. List the fields in the TCP header that are not part of the UDP header. Give the reason for each missing field.
33. An IP datagram is carrying a TCP segment destined for address 130.14.16.17. The destination port address is corrupted and it arrives at destination 130.14.16.19. How does the receiving TCP react to this error?
34. One ICMP message, discussed in Chapter 9, reports a destination port unreachable error. How can TCP detect the error in the destination port?
35. UDP is a message-oriented protocol. TCP is a byte-oriented protocol. If an application needs to protect the boundaries of its message, which protocol should be used, UDP or TCP?
36. What is the maximum size of the TCP header? What is the minimum size of the TCP header?
37. If the value of HLEN is 0111, how many bytes of option are included in the segment?
38. Show the entries for the header of a TCP segment that carries a message from an FTP client to an FTP server. Fill the checksum field with 0s. Choose an appropriate ephemeral port number and the correct well-known port number. The length of data is 40 bytes.
39. Write a short program to test if your computer is using big-endian or little-endian byte order.
40. What is the minimum length of a DHCP packet? What is the maximum length?
41. A DHCP packet is encapsulated in a UDP packet, which is encapsulated in an IP packet, which is encapsulated in an Ethernet frame. Find the efficiency of a DHCP packet when no option is used. The efficiency in this case is measured in the number of bytes in the DHCP packet to the total number of bytes transmitted at the data link layer.
42. Show an example of a DHCP packet with a padding option.

43. Show an example of a DHCP packet with an end-of-list option.
44. What is the maximum number of seconds that can be stored in the Number of Seconds field of a DHCP packet?
45. What is the size of a question record containing the domain name fhda.edu?
46. What is the size of a query message requesting the domain name for 185.34.23.12?
47. What do you think would happen if the control connection is accidentally severed during an FTP transfer?
48. Why should there be limitations on anonymous FTP? What could an unscrupulous user do?
49. A sender sends unformatted text. Show the MIME header.
50. A non-ASCII message of 1,000 bytes is encoded using base64. How many bytes are in the encoded message? How many bytes are redundant? What is the ratio of redundant bytes to the total message?
51. Do you think H.323 is actually the same as SIP? What are the differences? Make a comparison between the two.
52. Can H.323 also be used for video?

## Lesson plan

### Unit - I

1. Introduction: a brief history: ARPANET, birth of the internet, transmission control protocol/internetworking protocol (TCP/IP ), MILNET, CSNET, NSFNET, ANSNET, the internet today, world wide web, time line, growth of the internet.
2. Protocols and standards: protocols, standards; standards organizations: standards creation committees, forums, regulatory agencies; internet standards: maturity levels, requirement levels;
3. Internet administration: internet society (ISOC), internet architecture board (IAB), internet engineering task force (IETF), internet research task force (IRTF) ,internet assigned numbers authority (IANA) and internet corporation for assigned names and numbers (ICANN),network information center (NIC).
4. The OSI model and the TCP/IP protocol suite: protocol layers: hierarchy, services; the OSI model: layered architecture, layer-to-layer communication, encapsulation, layers in the OSI model, summary of OSI layers, TCP/IP protocol suite: comparison between OSI and TCP/IP protocol suite, layers in the TCP/IP protocol suite;
5. Introduction to Network Layer: introduction, switching, packet switching at network layer, network layer services.
6. Other network layer issues; ipv4 addresses: introduction, classful addressing.
7. Classless addressing, special addresses.
8. NAT, delivery and forwarding of ip packets: delivery.
9. Forwarding, structure of a router

### Unit - II

1. Internet protocol version 4 (ipv4), introduction datagram, fragmentation,
2. Options Checksum, Address Resolution Protocol (ARP): address mapping, the ARP protocol.
3. Mobile IP: addressing, agents, three phases, inefficiency in mobile IP
4. Unicast Routing Protocols: introduction, RIP,OSPF
5. BGP, multicasting and multicast routing protocols: introduction, multicast addresses.
6. IGMP
7. Introduction, Virtual Circuit and Datagram
8. Networks, Internet Protocol (IP), Routing algorithms,
9. Link state routing algorithms,
10. Distance vector routing algorithms,
11. Hierarchical routing, Routing in the Internet, Broadcast and Multicast Routing

**Unit - III**

1. Introduction to the transport layer: transport-layer services, transport-layer protocols
2. User Datagram Protocol (UDP): introduction, user datagram , UDP services,
3. UDP applications, Transmission Control Protocol (TCP): TCP services, TCP features, segment
4. A TCP connection, state transition diagram
5. Windows in TCP, flow control, error control,
6. Congestion control TCP timers
7. Stream Control Transmission Protocol (SCTP) : introduction, SCTP services , SCTP features.
8. Packet format, an SCTP association
9. State transition diagram, flow control.
10. Error control, congestion control.

**Unit -IV**

1. Introduction to the application layer: client-server paradigm, peer-to-peer paradigm; host configuration: DHCP: introduction.
2. DHCP operation, configuration, domain name system (DNS);
3. Need for DNS, name space, DNS in the internet.
4. Resolution, DNS messages, types of record.
5. Remote login: telnet and SSH: telnet, secure shell (SSH).
6. FTP, TFTP.
7. World wide web: architecture, web documents.
8. HTTP, electronic mail: architecture.
9. User agent, message transfer agent: SMTP, POP
10. Imap4, mime.
11. Multimedia: introduction, digitizing audio and video, streaming stored audio/video.
12. Streaming live audio/video, voice over IP, quality of service.

**Unit - V**

1. Ipv6 addressing :introduction, address space allocation,
2. Global unicast addresses, auto configuration,
3. Renumbering; IPv6 protocol: introduction, packet format, transition from ipv4 to IPv6;
4. Cryptography and network security: introduction, traditional ciphers,
5. Modern ciphers, asymmetric-key ciphers,
6. Message integrity, message authentication,
7. Digital signature, entity authentication, key management;
8. Internet security: network layer security,
9. Transport layer security,
10. Application layer security, firewalls

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Describes** different internet standards | **L1** | M | - | L | | M | | | | | M | H | | |
| **Compare** various routing algorithms like distance vector, link state, hierarchical & multicast routing, and understand the concept of fragmentation. **Use** different protocols to achieve Address mapping, Error reporting, unicast routing and multicasting | **L2 L3** | H | M | H | | M | | | | | | | M | L |
| **Differentiate** between TCP & UDP, and **understand** the services provided by them like flow control, congestion control etc. | **L4 L1** | M | L | H | | M | | | | | L | H | H | M |
| **Understand** and **recognize** the importance of application layer and the working of protocols like HTTP, FTP, DNS. **Identify** the different types of multimedia and **analyze** the quality of service provided by them | **L1 L6 L4** | M | L | H | | M | | | | | L | H | M | M |
| **Identify** and **solve** the problems associated with transition from IPV4 to IPV6. **Understand** the factors driving the need for network security. | **L3 L4** | H | M | L | | M | | | | | | | M | M |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcomes** | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Describes** different internet standards | **L1** | 2 | - | 1 | | 2 | | | | | 2 | 3 | | |
| **Compare** various routing algorithms like distance vector, link state, hierarchical & multicast routing, and understand the concept of fragmentation. **Use** different protocols to achieve Address mapping, Error reporting, unicast routing and multicasting | **L2 L3** | 3 | 2 | 3 | | 2 | | | | | | | 2 | 1 |
| **Differentiate** between TCP & UDP, and **understand** the services provided by them like flow control, congestion control etc. | **L4 L1** | 2 | 1 | 3 | | 2 | | | | | 1 | 3 | 3 | 2 |
| **Understand** and **recognize** the importance of application layer and the working of protocols like HTTP, FTP, DNS. **Identify** the different types of multimedia and **analyze** the quality of service provided by them | **L1 L6 L4** | 2 | 1 | 3 | | 2 | | | | | 1 | 3 | 2 | 2 |
| **Identify** and **solve** the problems associated with transition from IPV4 to IPV6. **Understand** the factors driving the need for network security. | **L3 L4** | 3 | 2 | 1 | | 2 | | | | | | | 2 | 2 |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |

| Course Title: Cloud Computing | | | |
|---|---|---|---|
| **Course Code: P13IS63** | **Semester : VI** | **L- T – P-H : 2 – 2 – 0-4** | **Credits: 3** |
| **Contact period : Lecture: 52 Hrs, Exam:3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |
| **Prerequisites:** Computer Network | | | |

## Course learning objectives(CLOs)

**This course aims to**

1. To learn how to use Cloud Services.
2. To implement Virtualization
3. To implement Task Scheduling algorithms.
4. Apply Map-Reduce concept to applications. And build Private Cloud.
5. To understand the concept of Virtualization and design of cloud Services.
6. To be familiar with the lead players in cloud.
7. To differentiate between full virtualization and para virtualization.
8. Works in teams to contribute, evaluate and feedback on case studies on different cloud computing solutions.
9. To know about security of virtualization.
10. To understand cloud Security Risks.

## Course Content

### Unit - I

**Introduction, Cloud Infrastructure**

Cloud computing, Cloud computing delivery models and services, Ethical issues, Cloud vulnerabilities, Cloud computing at Amazon, Cloud computing the Google perspective, Microsoft Windows Azure and online services, Open-source software platforms for private clouds, Cloud storage diversity and vendor lock-in, Energy use and ecological impact, Service level agreements, User experience and software licensing. **10 Hrs**

### Unit - II

**Cloud Computing: Application Paradigms.**

Challenges of cloud computing, Architectural styles of cloud computing, Workflows: Coordination of multiple activities, Coordination based on a state machine model: The Zookeeper, The Map Reduce programming model, A case study: The GrepTheWeb application , Cloud for science and engineering, High-performance computing on a cloud, Cloud computing for Biology research, Social computing, digital content and cloud computing. **10 Hrs**

### Unit- III

**Cloud Resource Virtualization.**

Virtualization, Layering and virtualization, Virtual machine monitors, Virtual Machines, Performance and Security Isolation, Full virtualization and para virtualization, Hardware support for virtualization, Case Study: Xen a VMM based para virtualization, Optimization of network virtualization, vBlades, Performance comparison of virtual machines, The dark side of virtualization. **10 Hrs**

### Unit- IV

**Cloud Resource Management and Scheduling.**

Policies and mechanisms for resource management, Application of control theory to task scheduling on a cloud, Stability of a two-level resource allocation architecture, Feedback control based on dynamic thresholds, Coordination of

specialized autonomic performance managers, A utility-based model for cloud-based Web services, Resourcing bundling: Combinatorial auctions for cloud resources, Scheduling algorithms for computing clouds, Fair queuing, Start-time fair queuing, Borrowed virtual time, Cloud scheduling subject to deadlines, Scheduling Map Reduce applications subject to deadlines, Resource management and dynamic scaling. **11 Hrs**

**Unit- V**

**Cloud Security, Cloud Application Development.**

Cloud security risks, Security: The top concern for cloud users, Privacy and privacy impact assessment, Trust, Operating system security, Virtual machine Security, Security of virtualization, Security risks posed by shared images, Security risks posed by a management OS, A trusted virtual machine monitor, Amazon web services: EC2 instances, Connecting clients to cloud instances through firewalls, Security rules for application and transport layer protocols in EC2, How to launch an EC2 Linux instance and connect to it, How to use S3 in java, Cloud-based simulation of a distributed trust algorithm, A trust management service, A cloud service for adaptive data streaming, Cloud based optimal FPGA synthesis . **11 Hrs**

**Text Book:**

1. Dan C Marinescu: Cloud Computing Theory and Practice. Elsevier(MK) 2013.

**Reference Books:**

1. Rajkumar Buyya , James Broberg, Andrzej Goscinski: Cloud Computing Principles and Paradigms, Willey 2014.
2. John W Rittinghouse, James F Ransome:Cloud Computing Implementation, Management and Security, CRC Press 2013.
3. Kumar Saurabh, "Cloud Computing – insights into New-Era Infrastructure", Wiley India, 2011.
4. Ronald L. Kurtz, Russell Dean Vines, "Cloud Security – A comprehensive Guide to Secure Cloud Computing", Wiley – India, 2010.

## Course Outcomes

**After learning all the units, the student is able to**

1. Demonstrate and experiment simple Cloud Applications and Compare the strengths and limitations of cloud computing
2. Apply resource allocation, scheduling algorithms and Implement Map-Reduce concept.
3. Create virtual machines from available physical resources and Familiarize with Open Stack.
4. Identify the architecture, infrastructure and delivery models of cloud computing
5. Apply suitable virtualization concept and Design Cloud Services and Set a private cloud

## Topic learning objectives

**After learning all the topics of unit-I, the student is able to**

1. Define cloud computing.-L1
2. List and describe different cloud computing models.-L1
3. Know about how to use cloud service.-L1
4. Explain the ethical issues of cloud computing.-L2
5. Discuss different cloud vulnerabilities.-L6
6. Describe cloud computing in Amazon and Google.-L1
7. Discuss the software platforms for private clouds.- L6
8. Understand cloud storage diversity and vendor lock-in.-L2
9. To know about energy use and ecological impact of cloud computing.-L1
10. Discuss the different service level agreements.-L6

**After learning all the topics of unit-II, the student is able to**

1. Describe the challenges of cloud computing-L1
2. Explain the architectural styles of cloud computing.-L2
3. Discuss the multiple activities coordination –L6
4. Understand the state machine model based on coordination.-L2
5. Define zookeeper.-L1
6. Discuss the map reduce programming model.-L6

7. Know about cloud for science and engineering.-L1
8. Describe the performance computing on a cloud.-L1
9. To know about use of cloud in biology research.-L1
10. To Know about digit content and cloud computing.-L1

**After learning all the topics of unit-III, the student is able to**

1. Define virtualization.-L1
2. Differentiate layering and virtualization.-L4
3. Describe the virtual monitors.-L1
4. To know about performance and security isolation.-L1
5. Differentiate between full virtualization and para virtualization.-L4
6. To know about hardware components that support virtualization.-L1
7. Discuss the xen a VMM based para virtualization.-L6
8. Describe the optimization of network virtualization.-L1
9. Discuss the virtualization blades.-L6
10. Explain the performance comparison of virtual machines-L2

**After learning all the topics of unit-IV, the student is able to**

1. To know about different policies and mechanisms for resource management.-L1
2. Explain the applications of control theory to task scheduling on a cloud.-L2
3. Discuss the stability of a two level resource allocation architecture.-L6
4. Understand dynamic thresholds based on feedback control.-L2
5. Describe the cloud based web service based on utility model.-L1
6. Discuss the combinatorial auctions for cloud resources.-L6
7. Explain scheduling algorithms for computing cloud.-L2
8. Describe different queuing methods in cloud resource management.-L1
9. Explain cloud scheduling subject to deadline.-L2
10. Discuss resource management and dynamic scaling.-L6

**After learning all the topics of unit-V, the student is able to**

1. To know about cloud security risks –L1
2. Describe top concern for cloud users.-L1
3. Define privacy. Explain privacy impact assessment.-L1. L2
4. Describe trust in cloud security.-L1
5. Differentiate between operating system security and virtual machine security.-L4
6. To know about security risks posed by shared images.-L1
7. To know about security risks posed by a management operating system.-L1
8. Explain the Amazon web services.-L2
9. Explain optimal FPGA synthesis base on cloud.-L2
10. Understand adaptive data streaming –L2

**Review questions**

1. Define Cloud Computing with example.
2. What are the properties of Cloud Computing? What is the working principle of Cloud Computing?
3. Explain the architecture of Cloud. Define Cloud services with example.
4. What are the advantages of cloud services?
5. Why is Cloud Computing important?
6. What are the advantages and disadvantages of Cloud Computing?
7. Who get benefits from Cloud Computing?
8. Who shouldn't be using Cloud Computing?

9. List the advantages and disadvantages of cloud service deployment.
10. List the companies who offer cloud service development?
11. What are the features of robust Cloud development? Who it offers?
12. Mention the Cloud service development tools.
13. Define the term web service with example.
14. What are the issues in web based applications?
15. Give the various schedules in Collaborating on schedule.
16. What is online collaboration in Collaborating on report?
17. What is difference between collaboration on task and event management?
18. What is virtual community?
19. List the enterprise-level web-based expense reporting Applications.
20. Explain the services provided by the Amazon infrastructure cloud from a user perspective
21. What is self service provisioning?
22. Explain a user view of Google App Engine with suitable block schematic.
23. Explain in brief, how cloud helps reducing capital expenditure?
24. What is the difference between process virtual machines, host VMMs and native VMMs
25. Explain some of the common pitfalls that come with virtualization.
26. What is the fundamental differences between the virtual machine as perceived by a traditional operating system processes and a system VM
27. Compare the SOAP and REST paradigms in the context of programmatic communication between applications deployed on different cloud providers, or between cloud applications and those deployed in –house
28. Explain the architecture of cloud file systems
29. Explain with suitable example, how a relational join could be executed in parallel using MapReduce.
30. Explain with suitable example the MapReduce model.
31. Why Cloud Computing brings new threats?
32. What is secure execution environment and communication in cloud? Explain different threats and vulnerabilities specific to virtual machines.
33. Explain the two fundamental functions, identity management and access control, which are required for secure cloud computing.
34. Explain risks from multi-tenancy, with respect to various cloud environments.
35. What is trusted cloud computing?
36. Explain issues in cloud computing with respect to implementing real time application over cloud platform.
37. Enlist and explain the principal design issues that are to be addressed while designing a QoS-aware distributed (middleware) architecture for cloud.
38. What is quality of service (QoS) monitoring in a cloud computing?
39. Enlist and explain different issues in inter-cloud environments.
40. Explain conceptual representation of the Eucalyptus Cloud.
41. Explain in brief the components within the Eucalyptus system
42. What is Nimbus?
43. What is the main way to deploy Nimbus Infrastructure?
44. What is the difference between cloudinit.d and the Context Broker?
45. What is Open Nebula Cloud? Explain main components of Open Nebula.
46. Explain Xen Cloud Platform (XCP) with suitable block diagram.
47. What are the collaboration schedules in communicating across the community?
48. Explain Collaborating on Group Projects and Events.
49. Explain the activities on cloud computing for the corporation?
50. Explain the mapping schedules managing projects.

## Lesson plan

### Unit - I

1. Cloud computing, Cloud computing delivery models and services.
2. Ethical issues, Cloud vulnerabilities.
3. Cloud computing at Amazon.
4. Cloud computing the Google perspective.
5. Microsoft Windows Azure and online services.
6. Open-source software platforms for private clouds.
7. Cloud storage diversity and vendor lock-in,
8. Energy use and ecological impact.
9. Service level agreements.
10. User experience and software licensing.

### Unit - II

1. Challenges of cloud computing, Architectural styles of cloud computing.
2. Workflows: Coordination of multiple activities.
3. Coordination based on a state machine model: The Zookeeper.
4. The Map Reduce programming model,
5. A case study: The GrepTheWeb application.
6. Cloud for science and engineering
7. High-performance computing on a cloud.
8. Cloud computing for Biology research.
9. Social computing.
10. Digital content and cloud computing.

### Unit- III

1. Virtualization, Layering and virtualization.
2. Virtual machine monitors, Virtual Machines.
3. Performance and Security Isolation.
4. Full virtualization and para virtualization.
5. Hardware support for virtualization.
6. Case Study: Xen a VMM based para virtualization.
7. Optimization of network virtualization.
8. VBlades.
9. Performance comparison of virtual machines.
10. The dark side of virtualization.

### Unit- IV

1. Policies and mechanisms for resource management, Application of control theory to task scheduling on a cloud.
2. Stability of a two-level resource allocation architecture.
3. Feedback control based on dynamic thresholds.
4. Coordination of specialized autonomic performance managers.
5. A utility-based model for cloud-based Web services.
6. Resourcing bundling: Combinatorial auctions for cloud resources.
7. Scheduling algorithms for computing clouds.
8. Fair queuing, Start-time fair queuing.
9. Borrowed virtual time, Cloud scheduling subject to deadlines.
10. Scheduling Map Reduce applications subject to deadlines.
11. Resource management and dynamic scaling.

### Unit- V

1. Cloud security risks, Security: The top concern for cloud users.
2. Privacy and privacy impact assessment, Trust.

3. Operating system security.
4. Virtual machine Security.
5. Security of virtualization.
6. Security risks posed by shared images, Security risks posed by a management OS.
7. A trusted virtual machine monitor.
8. Amazon web services: EC2 instances, Connecting clients to cloud instances through firewalls.
9. Security rules for application and transport layer protocols in EC2, How to launch an EC2 Linux instance and connect to it.
10. How to use S3 in java, Cloud-based simulation of a distributed trust algorithm.
11. A trust management service, A cloud service for adaptive data streaming, Cloud based optimal FPGA synthesis.

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | | |
| | | a | b | c | d | e | f | g | h | i | j | k | l | m |
| Demonstrate and experiment simple Cloud Applications and Compare the strengths and limitations of cloud computing | **L3 L6** | H | - | M | M | M | - | M | H | - | - | H | M | H |
| Apply resource allocation, scheduling algorithms and Implement Map-Reduce concept. | **L3 L5** | H | H | M | H | - | - | M | H | - | - | H | M | H |
| Create virtual machines from available physical resources and Familiarize with Open Stack. | **L5** | H | - | H | - | M | - | - | - | - | - | H | H | M |
| Identify the architecture, infrastructure and delivery models of cloud computing | **L1** | M | - | H | - | - | - | - | - | - | L | H | - | - |
| Apply suitable virtualization concept and Design Cloud Services and Set a private cloud | **L3 L5** | H | H | - | - | M | - | - | - | - | M | H | M | H |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |

| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | | |
| | | a | b | c | d | e | f | g | h | i | j | k | l | m |
| Demonstrate and experiment simple Cloud Applications and Compare the strengths and limitations of cloud computing | **L3 L6** | 3 | - | 2 | 2 | 2 | - | 2 | 3 | - | - | 3 | 2 | 3 |
| Apply resource allocation, scheduling algorithms and Implement Map-Reduce concept. | **L3 L5** | 3 | 3 | 2 | 3 | - | - | 2 | 3 | - | - | 3 | 2 | 3 |
| Create virtual machines from available physical resources and Familiarize with Open Stack. | **L5** | 3 | - | 3 | - | 2 | - | - | - | - | - | 3 | 3 | 2 |
| Identify the architecture, infrastructure and delivery models of cloud computing | **L1** | 2 | - | 3 | - | - | - | - | - | - | 1 | 3 | - | - |
| Apply suitable virtualization concept and Design Cloud Services and Set a private cloud | **L3 L5** | 3 | 3 | - | - | 2 | - | - | - | - | 2 | 3 | 2 | 3 |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |

| Course Title: C# & .NET | | | |
|---|---|---|---|
| **Course Code: P13IS64** | **Semester : VI** | **L- T – P -H: 4 – 0 – 0-4** | **Credits:4** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites :** Object Oriented Programming with Java

## Course Learning Objectives (CLOs)

**This course aims to**

1. Identify the basic building blocks of .NET.
2. Provide the concepts of fundamentals of Object oriented programming.
3. Implement the C# programming and .NET features.
4. Build C# Applications with and without using IDE.
5. Design the C# applications to solve real world problems.
6. Analyze the real world problems to solve using object-oriented approach.
7. Learn C# Language Fundamentals, Exception handling and life time of the objects.
8. Understand Interfaces and Collections in C#.
9. Understanding .NET Assembles and Building a Simple File Test Assembly.
10. Implement the advanced concepts of C# programming.

## Course Content

### Unit – I

**The Philosophy of .Net:** Understanding the Previous State of Affairs, The .NET Solution, The Building Block of the .NET Platform (CLR, CTS, and CLS), The Role of the .NET Base Class Libraries, What C# Brings to the Table, An Overview of .NET Binaries ( aka Assemblies ), The Role of the Common Intermediate Language , The Role of .NET Type Metadata, The Role of the Assembly Manifest, Compiling CIL to Platform – Specific Instructions, Understanding the Common Type System, Intrinsic CTS Data Types, Understanding the Common Languages Specification, Understanding the Common Language Runtime, A tour of the .NET Namespaces, Increasing Your Namespace Nomenclature, Deploying the .NET Runtime.

**Building C# Applications:** The Role of the Command Line Complier (csc.exe), Building C # Application using csc.exe, Working with csc.exe Response Files,

Generating Bug Reports , Remaining C# Compiler Options, The Command Line Debugger (cordbg.exe), Using the Visual Studio .NET IDE, Other Key Aspects of the VS.NET IDE, C# "Preprocessor:" Directives, An Interesting Aside: The System Environment Class. **10 Hrs**

### Unit – II

**C# Language Fundamentals:** The Anatomy of a Basic C# Class, Creating objects: Constructor Basics, The Composition of a C# Application, Default Assignment and Variable Scope, The C# Member Initialization Syntax, Basic Input and Output with the Console Class, Understanding Value Types and Reference Types, The Master Node: System.Object, The System Data Types (and C# Aliases), Converting Between Value Types and Reference Types: Boxing and Unboxing, Defining Program Constants, C# Iteration Constructs, C# Controls Flow Constructs, The Complete Set of C# Operators, Defining Custom Class Methods, Understating Static Methods, Methods Parameter Modifies, Array Manipulation in C #, String Manipulation in C#, C# Enumerations, Defining Structures in C#, Defining Custom Namespaces.

**Object- Oriented Programming With C#-1:** Forms Defining of the C# Class, Definition the "Default Public Interface" of a Type, Recapping the Pillars of OOP, The First Pillars: C#'s Encapsulation Services, Pseudo- Encapsulation: Creating Read-Only Fields. **11 Hrs**

### Unit – III

**Object- Oriented Programming With C#-2:** The Second Pillar: C#'s Inheritance Supports, keeping Family Secrets: The "Protected" Keyword, Nested Type Definitions, The Third Pillar: C #'s

Polymorphic Support, Casting Between.

**Exceptions And Object Lifetime:** Ode to Errors, Bugs, and Exceptions, The Role of .NET Exception Handing, The System. Exception Base Class, Throwing a Generic Exception, Catching Exception, CLR System – Level Exception (System. System Exception), Custom Application- Level Exception (System. System Exception), Handling Multiple Exception, The Family Block, the Last Chance Exception Dynamically Identifying Application – and System Level Exception Debugging System Exception Using VS. NET, Understanding Object Lifetime, the CIT of "new', The Basics of Garbage Collection, Finalization a Type, The Finalization Process, Building an Ad Hoc Destruction Method, Garbage Collection Optimizations, The System. GC Type.

**Interfaces:** Defining Interfaces Using C# Invoking Interface Members at the object Level, Exercising the Shapes Hierarchy, Understanding Explicit Interface Implementation, Interfaces As Polymorphic Agents, Building Interface Hierarchies, Implementing, Implementation, Interfaces Using VS .NET, understanding the IConvertible Interface, Building a Custom Enumerator (IEnumerable and Enumerator), Building Cloneable objects (ICloneable), Building Comparable Objects (I Comparable)

**11 Hrs**

**Unit – IV**

**Collections:** Exploring the System.Collections Namespace, Building a Custom Container (Retrofitting the Cars Type).

**Callback interfaces, delegates, events, and Advanced techniques-1:** Callback Interfaces, Understanding the .NET Delegate Type, Members of System. Multicast Delegate, The Simplest Possible Delegate Example, Building More a Elaborate Delegate Example, Understanding Asynchronous Delegates, Understanding (and Using) Events., The Advances Keywords of C#, A Catalog of C# Keywords Building a Custom Indexer, A Variation of the Cars Indexer Internal Representation of Type Indexer. Using C# Indexer from VB .NET, Overloading operators, The Internal Representation of Overloading Operators, Interacting with Overload Operator from Overloaded- Operator- Challenged Languages. **10 Hrs**

**Unit – V**

**ADVANCED TECHNIQUES-2:** Creating Custom Conversion Routines, Defining Implicit Conversion Routines, The Internal Representations of Customs Conversion Routines.

**UNDERSTANDING .NET ASSEMBLES:** Problems with Classic COM Binaries, An Overview of .NET Assembly, Building a Simple File Test Assembly, A C#. Client Application, A Visual Basic .NET Client Application, Cross Language Inheritance, Exploring the CarLibrary's, Manifest, Exploring the CarLibrary's Types, Building the Multifile Assembly, Using Assembly, Understanding Private Assemblies, Probing for Private Assemblies (The Basics), Private A Assemblies XML Configurations Files, Probing for Private Assemblies ( The Details), Understanding Shared Assembly, Understanding Shared Names, Building a Shared Assembly, Understanding Delay Signing, Installing/Removing Shared Assembly, Using a Shared Assembly. **10 Hrs**

**Text Books:**

1. Andrew Troelsen: Pro C# with .NET 3.0, Special Edition, Dream tech Press, India, 2007.
Chapters: 1 to 11 (Up to pp.369).
2. E Balaguruswamy: Programming in C#, 3rd edition, Tata McGraw Hill, Reprint 2014.

**References:**

1. Tom Archer: Inside C#, WP Publishers, 2007.
2. Herbert Schildt: C# The Complete Reference, Tata McGraw Hill.

## Course Outcomes

**After learning all the units of the course, the student is able to**

1. Understand conceptual framework of .Net environment.
2. Create Configuration for a given machine to host the .NET runtime.
3. Design .NET assemblies.
4. Design the C# program with well understanding of C# language constructs.
5. Conduct experiments with object oriented concepts in C# to solve real world problems.

## Topic Learning Objectives

**After learning all the topics of unit – I, the student is able to**

1. Differentiate between C# and Java – L3.
2. Is it possible to have two Main ( ) in a C# code? If so, how is it resolved without using an IDE? Explain with an example – L2.
3. Define assembly. Explain each component of an assembly – L1, L2.
4. Differentiate between single file assembly and multi file assembly – L3.
5. Explain the role of CIL – L2.
6. Explain the following: i) #region, #endregion ii) conditional code compilation –L2.
7. Explain the role of noconfig and debug flag – L2.
8. Explain the role of csc.rsp with an example – L2.
9. Explain the role of cordbg.exe by using its flags with an example –L2.
10. Write a C# program to display the following information using the System environment class: i) Current directory of application ii) Operating System Version – L3.

**After learning all the topics of unit – II, the student is able to**

1. List and explain any three instance methods and static methods of System.Object – L1, L2.
2. With an example, explain what happens when reference type is passed by value and passed by reference – L2.
3. Explain boxing and unboxing, with an example – L2.
4. Explain four method parameter modifiers, with an example – L2.
5. Write a C# program to arrange 5 names in ascending order. The names are obtained from the command line arguments – L3.
6. Write a program in C# to read a jagged array and display the sum of all the elements of three inner arrays – L3.
7. Differentiate between System.String and System.Text.StringBuilder – L3.
8. Write a C# program to demonstrate use of static and Read- only variables – L3.
9. Illustrate the use of static constructor with a program – L3.
10. Define Properties. Differentiate between Readonly and Writeonly Properties – L1, L3.

**After learning all the topics of unit – III, the student is able to**

1. Define Inheritance. Give Example. Discuss how the base class can be controlled using base – L1, L2.
2. Define Polymorphism. Explain how the Polymorphism can be enforced using abstract classes and abstract methods with one programming example – L1, L2, and L3.
3. Define Errors, Bugs and Exceptions. Discuss what is the role of .NET Exception Handling? – L1, L2.
4. List and explain with code, the core members of System.Exception type – L1, L2.
5. Define a method that would sort an array of integers. Incorporate exception handling mechanism for “index out of bounds” situation. Develop a main program that employs this method to sort a given set of integers – L3.
6. Describe the role of .NET garbage collection, finalization process and AdHoc destruction method with examples – L2.
7. Explain how garbage collection is optimized in .NET – L2.

8. Explain the need of interfaces – L2.
9. Differentiate between interfaces and abstract class – L3.
10. Explain with an example, explicit interface implementation – L2.

**After learning all the topics of unit – IV, the student is able to**

1. Explain the interfaces of System. Collection – L2.
2. Explain the need of Callback Interfaces – L2.
3. Differentiate between synchronous and asynchronous delegate, with examples – L3.
4. Write a program to illustrate the implementation of multicast delegates – L3.
5. Write a program to illustrate the implementation of event handler – L3.
6. List out the applications of delegates – L1.
7. Explain the need of event handling – L2.
8. Explain the need of operator overloading – L2.
9. List out the operators that can be overloaded – L1.
10. Write a program to illustrate the overloading of +, - and * operators – L3.

**After learning all the topics of unit – V, the student is able to**

1. Write a program to illustrate the use of Custom Conversion Routines – L3
2. Explain the need of an assembly – L2.
3. List out the problems of Classic COM Binaries – L1.
4. Explain the steps involved in building a single file assembly, with an example – L2.
5. With a program illustrate the C# support for cross language inheritance – L3.
6. Explain the steps involved in building a multi file assembly, with an example – L2.
7. Differentiate between single file assembly and multi file assembly – L3.
8. Explain the specific steps involved in binding to a private assembly at runtime – L2
9. Explain the need of a shared assembly – L2.
10. With a program illustrate the creation of a shared assembly – L3.

## Review Questions

1 Differentiate between C++ and C#.
2 Discuss the Core Features of .NET
3 Discuss the role of CLR, CTS, CLS and Base Class Libraries with the help of block diagram.
4 Discuss the Features of C# Language.
5 Define Metadata and Manifest.
6 With a neat diagram explain the working of Microsoft Common Object runtime execution engine (mscoree.dll).
7 Explain how to Create and Compile a C# program in Command Mode and .NET IDE Mode.
8 Explain how to refer external single and Multiple assemblies in command mode and .NET IDE mode.
9 Discuss how to compile the Multiple source files in Command Mode and .NET IDE Mode.
10 Explain the following aspects of VS.NET IDE :
i) Solution Explorer ii) Running and debugging iii) Documenting code via XML.
11 Give the Anatomy of a Basic C# Class. Discuss the different types of Main method.
12 Discuss the different types of Basic Input and Output with Console Class.
13 Differentiate between the Value types and reference types.
14 Discuss the characteristics of Static Method and Static data with examples.
15 Define Master Node. Explain the members of System. Object with examples.
16 Explain the following : i) for each loop ii) for loop
17 Differentiate between structure and enum.
18 Write a program in C# to sort an array of student objects having rollno, name and marks in two subjects.
-Display the array sorted on names.

-Display the array based on average marks.

19 Write a program in C# to accept two strings and perform the following operations:
i) Copy string 2 to string 3.
ii) Check string 1 ends with "ENGG" or not. If it is true, search character 'a' in string 3.

20 Discuss with illustration how the Encapsulation can be enforced using Traditional Accessors/Mutators and Class Properties.

21 Discuss the use of sealed classes with example.

22 Illustrate polymorphism with an example.

23 List and explain the core members of System. Exception type. How would you build custom exception?

24 Explain the keywords: i) finally ii) using.

25 Define object lifetime. Explain the Garbage Collection optimization process in C#.

26 Explain the process of finalizing objects in .NET environment.

27 Given the members of System. GC and explain their usage, with examples.

28 Write C# application to illustrate handling multiple exceptions.

29 What is the alternate approach to support multiple inheritance? List its major features.

30 Write a C# program which contains the following:
• An interface called dimension with the methods length( ) and width( ), which returns length and width in centimeters.
• Another interface called metric dimension with the methods length inches( ) and width inches( ),which returns length and width in inches.
• A class box that implements both the above said interfaces. This class has two data members length inches and width inches.
Define appropriate constructor for the class box. Write a main program to create an instance of box and to display the box length and width in inches and centimeters by invoking the appropriate methods of two interfaces.

31 With an example, explain any four interfaces of System. Collection.

32 With a program illustrate the use of callback interfaces.

33 Define delegate and explain its use with an example.

34 List out the steps involved in creating and using a delegate.

35 List out the special features of a delegate.

36 Differentiate between synchronous and asynchronous delegate, with examples.

37 Describe the syntax of an event declaration.

38 List out two typical examples where events are used.

39 Explain the need of operator overloading.

40 List out the operators that are not overloadable.

41 Explain the internal representations of customs conversion routines.

42 Explain the need of an assembly.

43 With a neat diagram, explain the physical view and logical view of. NET assemblies.

44 Explain the C# support for cross language inheritance, with examples.

45 Explain the steps involved in building a single file assembly, with an example.

46 Explain the steps involved in building a multi file assembly, with an example.

47 List out the differences between single file assembly and multi file assembly.

48 Explain the the specific steps involved in binding to a private assembly at runtime

49 With a program explain how we can able to build a shared assembly.

50 With a program explain how we can install/remove a shared assembly.

## Lesson Plan

### Unit I

1 **The Philosophy of .Net:** Understanding the Previous State of Affairs, The .NET Solution
2 The Building Block of the .NET Platform (CLR, CTS, and CLS), The Role of the .NET Base Class Libraries, What C# Brings to the Table,
3 An Overview of .NET Binaries ( aka Assemblies ), The Role of the Common Intermediate Language , The Role of .NET Type Metadata
4 The Role of the Assembly Manifest, Compiling CIL to Platform – Specific Instructions, Understanding the Common Type System
5 Intrinsic CTS Data Types, Understanding the Common Languages Specification, Understanding the Common Language Runtime
6 A tour of the .NET Namespaces, Increasing Your Namespace Nomenclature, Deploying the .NET Runtime.
7 **Building C# Applications:** The Role of the Command Line Complier (csc.exe), Building C # Application using csc.exe
8 Working with csc.exe Response Files, Generating Bug Reports , Remaining C# Compiler Options,
9 The Command Line Debugger (cordbg.exe) Using the, Visual Studio .NET IDE, Other Key Aspects of the VS.NET IDE
10 C# "Preprocessor:" Directives, An Interesting Aside: The System Environment Class.

### Unit II

1 **C# Language Fundamentals:** The Anatomy of a Basic C# Class, Creating objects: Constructor Basics
2 The Composition of a C# Application, Default Assignment and Variable Scope, The C# Member Initialization Syntax
3 Basic Input and Output with the Console Class, Understanding Value Types and Reference Types
4 The Master Node: System, Object, The System Data Types (and C# Aliases)
5 Converting Between Value Types and Reference Types: Boxing and Unboxing, Defining Program Constants, C# Iteration Constructs, C# Controls Flow Constructs
6 The Complete Set of C# Operators, Defining Custom Class Methods, Understating Static Methods, Methods Parameter Modifies
7 Array Manipulation in C #, String Manipulation in C#
8 C# Enumerations, Defining Structures in C#, Defining Custom Namespaces.
9 **Object- Oriented Programming With C#-1:** Forms Defining of the C# Class
10 Definition the "Default Public Interface" of a Type, Recapping the Pillars of OOP
11 The First Pillars: C#'s Encapsulation Services, Pseudo- Encapsulation: Creating Read-Only Fields.

### Unit III

1 **Object- Oriented Programming With C#-2:** The Second Pillar: C#'s Inheritance Supports, keeping Family Secrets: The "Protected" Keyword, Nested Type Definitions
2 The Third Pillar: C #'s Polymorphic Support, Casting Between.
3 **Exceptions And Object Lifetime:** Ode to Errors, Bugs, and Exceptions, The Role of .NET Exception Handing
4 The System. Exception Base Class, Throwing a Generic Exception, Catching Exception, CLR System – Level Exception (System. System Exception)
5 Custom Application- Level Exception (System. System Exception), Handling Multiple Exception, The Finally Block

6 The Last Chance Exception Dynamically Identifying Application – and System Level Exception Debugging System Exception Using VS. NET, Understanding Object Lifetime, the CIT of "new'
7 The Basics of Garbage Collection, Finalization a Type, The Finalization Process, Building an Ad Hoc Destruction Method, Garbage Collection Optimizations, The System. GC Type.
8 **Interfaces:** Defining Interfaces Using C# Invoking Interface Members at the object Level
9 Exercising the Shapes Hierarchy, Understanding Explicit Interface Implementation, Interfaces As Polymorphic Agents
10 Building Interface Hierarchies, Implementing, Implementation, Interfaces Using VS .NET, understanding the IConvertible Interface
11 Building a Custom Enumerator (IEnumerable and Enumerator), Building Cloneable objects (ICloneable), Building Comparable Objects (I Comparable)

**Unit IV**

1 **Collections:** Exploring the system. Collections Namespace
2 Building a Custom Container (Retrofitting the Cars Type).
3 **Callback interfaces, delegates, and events, Advanced techniques-1:** Callback Interfaces, Understanding the .NET Delegate Type
4 Members of System. Multicast Delegate, The Simplest Possible Delegate Example
5 Building More a Elaborate Delegate Example, Understanding Asynchronous Delegates
6 Understanding (and Using) Events., The Advances Keywords of C#
7 A Catalog of C# Keywords Building a Custom Indexer
8 A Variation of the Cars Indexer Internal Representation of Type Indexer. Using C# Indexer from VB .NET
9 Overloading operators, The Internal Representation of Overloading Operators
10 Interacting with Overload Operator from Overloaded- Operator- Challenged Languages,

**Unit V**

1 **Advanced Techniques-2:** Creating Custom Conversion Routines, Defining Implicit Conversion Routines
2 The Internal Representations of Customs Conversion Routines.
3 **Understanding .Net Assembles:** Problems with Classic COM Binaries
4 An Overview of .NET Assembly, Building a Simple File Test Assembly
5 A C#. Client Application, A Visual Basic .NET Client Application
6 Cross Language Inheritance, Exploring the CarLibrary's, Manifest, Exploring the CarLibrary's Types
7 Building the Multifile Assembly, Using Assembly, Understanding Private Assemblies, Probing for Private Assemblies (The Basics)
8 Private A Assemblies XML Configurations Files, Probing for Private Assemblies ( The Details)
9 Understanding Shared Assembly, Understanding Shared Names, Building a Shared Assembly
10 Understanding Delay Signing, Installing/Removing Shared Assembly, Using a Shared Assembly.

| Course Articulation Matrix (CAM) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome (ABET/NBA-(3a-m))** | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Understand conceptual framework of .Net environment. | **L2** | | | | | | | | | M | | | | L |
| Create Configuration for a given machine to host the .NET runtime. | **L2** | | | | | | | | | M | | M | | |
| Design .NET assemblies. | **L3** | | | | | | | | | M | | M | | |
| Design the C# program with well understanding of C# language constructs. | **L3** | L | M | M | M | L | | | | M | | M | | |
| Conduct experiments with object oriented concepts in C# to solve real world problems. | **L4** | M | | M | M | M | M | | | M | | H | M | L |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcome (CO)** | | **Program Outcome(ABET/NBA-(3a-m))** | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Understand conceptual framework of .Net environment. | **L2** | | | | | | | | | 2 | | | | 1 |
| Create Configuration for a given machine to host the .NET runtime. | **L2** | | | | | | | | | 2 | | 2 | | |
| Design .NET assemblies. | **L3** | | | | | | | | | 2 | | 2 | | |
| Design the C# program with well understanding of C# language constructs. | **L3** | 1 | 2 | 2 | 2 | 1 | | | | 2 | | 2 | | |
| Conduct experiments with object oriented concepts in C# to solve real world problems. | **L4** | 2 | | 2 | 2 | 2 | 2 | | | 2 | | 3 | 2 | 1 |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |

| **Course Title: Modern Information Retrieval** | | | |
|---|---|---|---|
| **Course Code: P13IS65** | **Semester : VI** | **L- T – P-H : 4 – 0 – 0-4** | **Credits:4** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites:** The student should have undergone the course on programming language.

## Course learning objectives

**This course aims to**

1. Understand the difficulty of representing and retrieving documents.
2. Understand the latest technologies for linking, describing and searching the Web.
3. Understand the relationship between IR, hypermedia, and semantic models.

4. Be familiar with classical techniques of Information Retrieval, and the additional techniques employed by Web search engines sufficient to understand how Web search engines work and how they could be improved.
5. Be familiar with techniques for conveying the meaning of documents or hypermedia content, for example, metadata, thesauri, and classification taxonomies – sufficient to understand their application to the "semantic Web".
6. Be familiar with the fundamentals of hypermedia systems sufficient to know how to develop a good Web hypermedia and why a Web site is good or bad.
7. Be able to implement techniques for the preprocessing needed for information retrieval systems.
8. Be able to develop a small information retrieval system.

## Course contents

### Unit-I

**Information Retrieval and Modeling**

Introduction: Information versus data retrieval, information retrieval at the center of stage; basic concepts: user task, logical view of the documents; past present and future: early development, information retrieval in the library, the web and digital libraries; the retrieval process.

**Modeling:** A taxonomy of IR models; classic IR: Basic Concepts, Boolean model, vector model, probabilistic model, fuzzy set model, extended Boolean model, neural network model. **10 Hrs**

### Unit-II

**Retrieval Evaluation, Query Language and Operation**

Introduction; retrieval performance evaluation, reference collection, keyword-based querying pattern matching, structural queries, query protocol, user relevance feedback, automatic local analysis, automatic global analysis, metadata, text, markup language, multimedia. **10 Hrs**

### Unit-III

**Text Operation, Indexing and Searching**

Introduction, document preprocessing, document clustering, text compression, inverted files, other indices for text, Boolean queries, sequential searching, pattern matching, structural queries, compression **10 Hrs**

### Unit-IV

**Parallel and Distributed IR and Multimedia IR**

Introduction, parallel IR, distributed IR, data modeling, query language, spatial access methods, a generic multimedia indexing approach, one dimensional time series, two dimensional color images. **12 Hrs**

### Unit-V

**Searching the Web**

Introduction, challenges, characterizing the web, search engines browsing, metasearchers, finding the needle in the haystack, searching using hyperlinks **10 Hrs**

**Text Books:**

1. Ricardo Baeza-Yates, Berthier Ribeiro-Neto: Modern Information Retrieval, First edition, Pearson Education, 2011.

**Reference Book:**

1. William B Frakes, Ricardo Baeza Yates: Information Retrieval Data Structures and Algorithms, First edition, Pearson Education, 2009.
2. C.J. van Rijsbergen: Information Retrieval, Second Edition, Butterworth-Heinemann.

## Course outcomes

**After learning all the units of the course, the student is able to**

1. Compare data & information retrieval systems and Illustrate various information retrieval models such as Boolean, vector, probabilistic models and describe file structures such as inverted files.
2. Describe Boolean queries and performance evaluation measures such as recall and precision.
3. Discuss the preprocessing, clustering and compression of text and understand the uses of inverted files and other indices for text.
4. Understand MIMD architectures and issues of distributed information retrieval such as collection partitioning, source selection and query processing and Explain data modeling, spatial access methods and multimedia indexing approach.
5. Explain the uses of Ranking and Crawling the Web in search engines.

## Topic learning objectives

**After learning all the topics of unit-I, the student is able to**

1. Define information retrieval. -L1
2. Distinguish information retrieval and Data Retrieval -L2
3. Know about past, Present and future of Information Retrieval.-L1
4. Discuss the Retrieval process.-L2
5. Discuss the Boolean model -L2
6. Understand the uses of vector model -L2
7. Understand the probalistic model -L2
8. Know about Fuzzy set model in IR modeling.-L1
9. Know about Extended Boolean Model in IR modeling. -L1

**After learning all the topics of unit-II, the student is able to**

1. Know about the meaning of retrieval evaluation. -L1
2. Understand the uses of Precision and Recall in performance evaluation.-L2
3. Discuss about TRCC, CACM and ISI collections. -L2
4. Know about the working of keyword based querying. -L1
5. Describe the pattern matching in query language. -L1
6. Explain about hypertext, hierarchical structure queries. -L2
7. Discuss the user relevant Feedback in query operation. -L2
8. Compare automatic local and automatic global analysis. -L6
9. Explain the uses of text, and markup languages in IR model. -L2

**After learning all the topics of unit-III, the student is able to**

1. Know about uses of text in IR model.-L2
2. Discuss the lexical analysis of the text in Document Preprocessing.-L2
3. Understand the stemming in document preprocessing. -L2
4. Know about the elimination of stopwords in document preprocessing. -L1
5. Know about document clustering. -L1
6. Discuss the basic concepts, statistical and dictionary methods in text compression. -L2
7. Discuss the uses of inverted files in IR model. -L2
8. Describe different mechanism are used for information searching. -L1
9. Explain the different methods of sequential searching. -L2

**After learning all the topics of unit-IV, the student is able to**

1. Understand the parallel computing in IR. -L2
2. Know about performance measure techniques. -L1
3. Discuss the MIMD architecture -L2

4. Discuss the SIMD architecture. -L2
5. Explain the working of Distributed IR methods. -L1
6. Describe the how multimedia data support in commercial DBMS. -L2
7. Discuss the conditions on Multimedia data. -L2
8. Understand the spatial access methods in Multimedia IR. -L2
9. Know about the one dimensional time series and two dimensional color images. -L1

**After learning all the topics of unit-V, the student is able to**

1. Discuss the different challenges of web searching. -L2
2. Understand the measuring and modeling the web. -L2
3. Compare centralized architecture and distributed architecture in search engine. -L6
4. Know about browsing operation. -L1
5. Define metasearcher. -L1
6. Know about the needle in the Haystack. -L1
7. Know about different web query languages. -L1
8. Discuss the working of static search and dynamic search. -L2
9. Understand the role of software agents. -L2

## Review Questions

1. How the Information Retrieval System is related to Database Management System?
2. Explain various types of Information Retrieval Systems.
3. Differentiate Full Item indexing, Public File Indexing and Private File Indexing.
4. Write short notes on Multimedia indexing
5. Write short notes on Indexing by Term
6. Write short notes on Indexing by concept.
7. Describe document and term clustering?
8. Write about the hierarchy of clusters?
9. Write about Selective dissemination of information search
10. Write about Searching the Internet
11. Write about Information visualization technologies.
12. Explain the various text searching algorithms.
13. Explain how information system evaluation is handled?
14. Write about different measures used in system evaluation.
15. Write about Multimedia searching techniques
16. Write about Digital libraries
17. Explain porter stemming algorithm?
18. What are the difference between document clustering and term clustering.
19. What is relevance feedback?
20. Explain the Rochio method for relevance feedback.
21. Discuss the data modeling in light of multimedia information Retrieval.
22. What is an indexing?
23. Explain the objectives of indexing?
24. Discuss about techniques involved in software text search algorithm.
25. Explain about Boolean model with example.
26. Explain about vector model with example.
27. Explain the probabilistic model in IR?
28. Describe the similarities and differences between stemming and N-grams
29. Describe the concept of information extraction.
30. Distinguish between Bayesian networks and inference network model.
31. Write a short note on ranking algorithms.

32. Write a short note on similarity measures.
33. How Bayesian networks are used in maintain information.
34. Compare and contrast the term clustering and item clustering.
35. Differentiate hardware versus software text search algorithms.
36. Write about boyers Moore search algorithm
37. Write about kunth mories search algorithm.
38. Differentiate between boyers-moore and kunth mories search algorithm.
39. Write a note MIMD architecture.
40. Write a note on SIMD architecture
41. Explain about query processing in distributed IR?
42. Explain about statistical and dictionary methods in text compression.
43. Write a note on markup languages.
44. Explain the formats of multimedia.
45. Distinguish between query expansion based on a similarity thesaurus and query expansion based on a statistical thesaurus
46. Distinguish between query expansion through local clustering and query expansion through local context analysis.
47. Explain context and Boolean queries.
48. Explain inverted file compression.
49. Explain the different models for browsing.
50. Define ranking. Explain how ranking used for search engines.
51. Write a short note on information retrieval system?
52. Explain the different text and multimedia languages and properties of Information Retrieval system

## Lesson Plan

### Unit-I

1. Introduction: motivation: information versus data retrieval, information retrieval at the center of the stage;
2. Basic concepts: the user task, logical view of the documents;
3. Past present and future: early development, information retrieval in the library, the Web and digital libraries; the retrieval process.
4. Modeling: A taxonomy of IR models; classic IR.
5. Basic Concepts, Boolean model,
6. Vector model,
7. Probabilistic model.
8. Fuzzy set model.
9. Extended Boolean model.
10. Neural network model.

### Unit-II

1. Introduction; retrieval performance evaluation.
2. Recall and precision and Alternative measures
3. Reference collection: TRCC collection.
4. CACM and ISI collection
5. Keyword-based querying
6. Pattern matching,
7. Structural queries, query protocol,
8. User relevance feedback,
9. Automatic local analysis, and automatic global analysis,
10. Metadata, text, markup language, multimedia.

**Unit-III**

1. Introduction, document preprocessing,
2. Document clustering: lexical analysis of the text.
3. Elimination of stop words, stemming, index terms selection, thesauri
4. Text compression,
5. Inverted files,
6. Other indices for text,
7. Boolean queries,
8. Sequential searching,
9. Pattern matching,
10. Structural queries, compression

**Unit-IV**

1. Introduction: parallel computing, performance measures
2. Parallel IR: MIMD architecture
3. SIMD architecture
4. Distributed IR,
5. Distributed IR (continue)
6. Data modeling,
7. Query language
8. Query language (continue)
9. Spatial access methods,
10. A generic multimedia indexing approach,
11. One dimensional time series,
12. Two dimensional color images.

**Unit-V**

1. Introduction, challenges,
2. Characterizing the web,
3. Characterizing the web( continue)
4. Search engines
5. Search engines(continue)
6. Browsing,
7. Metasearchers,
8. Finding the needle in the haystack,
9. Searching using hyperlinks: web query language
10. Dynamic search and software agents

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Compare** data & information retrieval systems and **Illustrate** various information retrieval models such as Boolean, vector, probabilistic models and describe file structures such as inverted files. | **L6 L4** | M | L | | | | | L | | L | M | H | M | |
| **Describe** Boolean queries and performance evaluation measures such as recall and precision. | **L1** | H | M | M | | | | M | | | M | M | M | |
| **Discuss** the preprocessing, clustering and compression of text and **understand** the uses of inverted files and other indices for text. | **L2** | M | L | | | | | | | | H | M | M | |
| **Understand** MIMD architectures and issues of distributed information retrieval such as collection partitioning, source selection and query processing and **Explain** data modeling, spatial access methods and multimedia indexing approach. | **L2 L1** | M | L | | | | | M | | | M | M | M | |
| **Explain** the uses of Ranking and Crawling the Web in search engines | **L1** | M | M | | | | | | | | M | L | M | |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |

| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome(ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Compare** data & information retrieval systems and **Illustrate** various information retrieval models such as Boolean, vector, probabilistic models and describe file structures such as inverted files. | **L6 L4** | 2 | 1 | | | | | 1 | | 1 | 2 | 3 | 2 | |
| **Describe** Boolean queries and performance evaluation measures such as recall and precision. | **L1** | 3 | 2 | 2 | | | | 2 | | | 2 | 2 | 2 | |
| **Discuss** the preprocessing, clustering and compression of text and **understand** the uses of inverted files and other indices for text. | **L2** | 2 | 1 | | | | | | | | 3 | 2 | 2 | |
| **Understand** MIMD architectures and issues of distributed information retrieval such as collection partitioning, source selection and query processing and **Explain** data modeling, spatial access methods and multimedia indexing approach. | **L2 L1** | 2 | 1 | | | | | 2 | | | 2 | 2 | 2 | |
| **Explain** the uses of Ranking and Crawling the Web in search engines | **L1** | 2 | 2 | | | | | | | | 2 | 1 | 2 | |
| **1 – Low, 2 – Moderate and 3 – High** | | | | | | | | | | | | | | |

| Course Title: C# & .NET LABORATORY | | | |
|---|---|---|---|
| **Course Code: P13ISL67** | **Semester : VI** | **L- T – P -H: 0 – 0 – 3-3** | **Credits1.5** |
| **Contact period : Lecture : 39 Hrs, Exam: 3Hrs** | | **Weightage: CIE: 50% ; SEE: 50%** | |

## Course Learning Objectives

**This course aims to**

1. Understand programming language concepts, particularly C# and object-oriented concepts.
2. Write and debug .NET applications.
3. Implement .NET classes from specifications.
4. Effectively create and use objects from class libraries.
5. Understand the behaviour of primitive data types, object references, and arrays.
6. Apply decision and iteration control structures to implement algorithms.
7. Write C# programs using simple recursive functions.
8. Implement interfaces, inheritance, and polymorphism as programming techniques.
9. Apply exceptions handling.
10. Build and use shared assemblies.

## Course Outcomes

**After learning all the units of the course, the student is able to**

1. Develop and debug C# programs with or without using IDE.
2. Create Configuration for a given machine to host the .NET runtime.
3. Design .NET assemblies.
4. Develop and debug the C# programs with well understanding of C# language constructs.
5. Conduct experiments with object oriented concepts in C# to solve real world problems.

## LAB EXPERIMENTS

1. Program to build C# application using csc.exe.
2. Generating bug reports and working with c# compiler options.
3. Experiments on C# language fundamentals.
4. Programs to implement basic object oriented features in C#.
5. Programs on exception handling and dealing with object lifetime.
6. Programs to implement interfaces in C#.
7. Programs to demonstrate usage of delegates.
8. Programs to demonstrate working of events.
9. Programs to build multifile assemblies.
10. Programs to build and use shared assemblies.

| Course Title: NETWORKING LABORATORY | | | |
|---|---|---|---|
| **Course Code: P13ISL68** | **Semester : VI** | **L- T – P-H : 0 – 0 – 3-3** | **Credits:1.5** |
| **Contact period : Lecture: 39 Hrs, Exam: 3 Hrs** | | **Weightage: CIE:50%; SEE:50%** | |

**Prerequisites:** Know about computer network, any Simulator working and C language

## Course Learning Objectives

**This course aims to**

1. Write, execute and debug c programs which use Socket API.
2. Understand the use of client/server architecture in application development
3. Understand how to use TCP and UDP based sockets and their differences.
4. Design reliable servers using both TCP and UDP sockets and analyze various protocol parameters of TCP/IP reference model.
5. Understand the working of simulators like OPNET+ / NS2.
6. Understand the working of packet sniffers like wireshark.
7. Learn how to use C/C++ code to implement network related concepts.
8. Design network models and analyze its performance by specifying parameters.

## Course Contents

1. Write a program for error detection using CRC-CCITT (16-bits).
2. Write a program to generate Hamming Code for error detection and correction.
3. Write a program for congestion control using Leaky bucket algorithm..
4. Write a program for distance vector algorithm to find suitable path for transmission.
5. Write a program for spanning tree algorithm to find loopless path in a network.
6. Trace Hyper Text Transfer Protocol.
7. Trace Domain Name Server.
8. Trace Transmission control protocol.
9. Trace Internet Protocol,
10. Trace Internet Control Message Protocol.
11. Trace Dynamic Host Configuration Protocol.
12. Trace ARP.
13. Simulate a three nodes point-to-point network with duplex links between them. Set the queue size vary the bandwidth and find the number of packets dropped.
14. Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 and n2-n3. Apply TCP agent between n0-n3 and UDP agent between n1-n3. Apply relevant applications over TCP and UDP agents by changing the parameters and determine the number of packets sent by TCP/UDP.
15. Simulate the different type of internet traffic such as FTP and TELNET over a network and analyze the throughput.
16. Simulate a transmission of ping message over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
17. Simulate an Ethernet LAN using n nodes, change error rate and data rate and compare the throughput.
18. Simulate an Ethernet LAN using n nodes and set multiple traffic nodes and determine the collision across different nodes.
19. Simulate an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source/destination
20. Simulate a wireless network and transfer UDP data on it.
21. Simulate a wireless network and understand the concept of Carrier-Sensing in IEEE 802.11
22. Simulate a wireless network and explore the impact of MAC overhead and multiple hops on achievable data throughput in a wireless network.

**Elective – I Group – A**

| Course Title: Storage Area Network | | | |
|---|---|---|---|
| **Course Code: P13ISL661** | **Semester : VI** | **L- T – P-H : 4 – 0 – 0-4** | **Credits: 4** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites :** Operating system and Communication Network.

**Course Learning Objectives (CLOs)**

**This course aims to**

1. Provide a comprehensive view of storage architectures, and the logical and
2. Physical components of storage infrastructure including storage subsystems
3. Discuss RAID and intelligent storage systems
4. Discuss storage networking technologies such as FC SAN, IP SAN, and FCoE
5. Discuss NAS, object-based and unified storage
6. Discuss business continuity, storage security, storage monitoring and management activities

**Course Content**

**Unit -I**

Introduction: Information Storage, Evolution of Storage Architecture, Data Centre Infrastructure, Virtualization and Cloud Computing.

Data Centre Environment: Application, DBMS, Host, Connectivity, Storage, Disk Drive Components, Disk Drive Performance, Host Access to Data, Direct-Attached Storage, Storage Design Based on Application, Disk Native Command Queuing, Introduction to Flash Drives. **10 Hrs**

**Unit -II**

Data Protection: RAID Implementation Methods, Array Components, Techniques, Levels, Impact on Disk Performance, Comparison, Hot Spares.

Intelligent Storage System: Components, Storage Provisioning, Types of intelligent storage systems. **10 Hrs**

**Unit- III**

Fibre Channel Storage Area Networks: FC Overview, Evolution, Components, FC Connectivity, Ports, FC Architecture, Fabric Services, Login Types, Zoning, FC Topologies, Virtualization in SAN. IP SAN and FCoE: iSCSI, FCIP, FCoE. **12 Hrs**

**Unit- IV**

Network-Attached Storage: Benefits, Components, NAS I/O Operation, Implementations, File Sharing Protocols, Factors Affecting NAS Performance, File-Level Virtualization.

Object Based and Unified Storage: Object Based Storage Devices, Content Addressed Storage, CAS Use Cases, Unified Storage. **10 Hrs**

**Unit- V**

Business Continuity: Information Availability, Terminology, Planning Lifecycle, Failure Analysis, Impact Analysis, Solutions. Cloud Computing Infrastructure.

Securing the Storage Infrastructure: Framework, Risk Triad, Domains. Security Implementations in Storage Networking, Virtualized and Cloud Environments.

Managing the Storage Infrastructure: Monitoring, Management Activities, Management Challenges, Information Lifecycle Management, Storage Tiering. **10 Hrs**

**Text Book:**

1. Somasundaram G., Alok Shrivastava, (EMC Education Services); "Information Storage and Management"; 2e, Wiley India, 2012, ISBN 9788126537501.

**Reference Book:**

1.Robert Spalding; "Storage Networks: The Complete Reference"; Tata McGraw Hill, 2003.

## Course Outcome

**At the end of the course the student should be able to**

1. Explain storage architectures, and the logical and physical components of storage infrastructure including storage subsystems.
2. Describe RAID and intelligent storage systems
3. Illustrate storage networking technologies such as FC SAN, IP SAN, and FcoE
4. Discuss NAS, and object-based and unified storage
5. Explain business continuity, storage security, storage monitoring and management activities

## Topic Learning Objectives

**After learning all the topics of unit – I, the student is able to**

1. Describe data and information.-L1
2. Differentiate structured data and unstructured data.-L4
3. Understand evolution of storage architecture.-L2
4. Discuss different core element of data center.-L6
5. Explain virtualization and cloud computing.-L2
6. Discuss applications of Virtualization.-L6
7. Distinguish b/w file system and volume manger.-L2
8. Discuss disk drive components.-L6
9. Explain logical block addressing.-L2

**After learning all the topics of unit – II, the student is able to**

1. List out hardware and software components of RAID.-L1
2. Discuss stripping ,mirroring and parity.-L6
3. Describe different RAID Levels.-L1
4. Explain RAID write penalty.-L2
5. Explain hot spares.-L2
6. Discuss components of an intelligent storage systems.-L6
7. Define storage provisioning.-L1
8. Differentiate high end storage system and mid range storage system .-L4

**After learning all the topics of unit – III, the student is able to**

1. Describe fiber channel architecture.-L1
2. Know about ports in Fiber channels SAN.-L1
3. Discuss Fibre Channel protocol stack.-L6
4. Explain Fibre channel addressing.-L2
5. Know about world wide names in SAN.-L1
6. Define Zoning.-L1
7. Distinguish b/w native iSCSI and Bridged iSCSI.-L2
8. Differentiate FCIP protocol and FcoE protocol.-L4

**After learning all the topics of unit – IV, the student is able to**

1. List out network-attached storage devices.-L1
2. Describe benefits of NAS.-L1
3. Discuss the performance of NAS.-L6
4. Explain NAS connectivity and protocols.-L2
5. Discuss file level virtualization.-L6
6. Distinguish b/w object based storage and content address strorage.-L2
7. Know about unified storage.-L1

**After learning all the topics of unit – V, the student is able to**

1. Know about disaster recovery in business continuity.-L1
2. Explain business impact analysis.-L2
3. Define business continuity planning.-L1
4. Discuss storage security frame work.-L6
5. Describe Denial of service in security.-L1
6. Know about management platform standrds.-L1
7. discuss Information life cycle management.-L6
8. Explain storage tiering.-L2
9. Explain monitoring and alerts in managing the storage infrastructure.-L2

## Review Questions

1. What is structured and unstructured data?
2. Research the challenges of storing and managing unstructured data.
3. Discuss the benefits of information-centric storage architecture over server centric architecture.
4. What are the attributes of big data ? Research and prepare a presentation on big data analytics.
5. Research how business use their information assets to derive competitive advantage and new business opportunities.
6. Prepare a presentation on personal data management.
7. What are the advantages of a virtualized data centre over a classic data centre.
8. Which components constitute the disk service time ? Which components contributes the largest percentage of the disk service time in a random I/O operation?
9. Research other elements of a data centre besides the core elements ,including environmental control parameters such a HVAC,power supplies, and security.
10. Why is RAID 1 not a substitute for a backup ?
11. Research RAID 6 and its second parity computation.
12. Explain the process of data recovery in case of a drive failure in RAID 5.
13. What are the benefits of using RAID 3 in a backup application?
14. Discuss the impact of random and sequential I/Os in different RAID configurations.
15. What is the stripe size of a five disk RAID 5 set with strip size of 32 KB?
16. Research cache Coherency mechanisms, and explain how they address the environment with multiple shared caches.
17. Which type of application benefits the most by bypassing write cache? Justify your answer.
18. Prepare a presentation on EMC VMAX architecture.
19. What is Zoning?
20. Describe the process of assigning an FC address to a node when logging on to the network for the first time.
21. Discuss the roles of the name server and fabric controller in an FC-switched fabric.
22. How does flow control work in an FC network?
23. Explain storage migration using block level storage virtualization.
24. How do VSANs improve the manageability of an FC SAN?
25. How does iSCSI handle the process of authentication ?
26. Why should an MTU value of at least 2,500 bytes be configured in a bridged iSCSI environment?
27. Why does the lossy nature of standard Ethernet make it unsuitable for a layered FCoE implementation?
28. Compare various data centre protocols that use Ethernet as the physical medium for transporting storage traffic.
29. Explain how the performance of NAS can be affected if the TCP window size at the sender and receiver are nor synchronized.
30. How does the use of jumbo frames affect the NAS performance?

31. Explain the file access and sharing features of pNFS.
32. How does file-Level virtualization ensure non disruptive file mobility?
33. Discuss the object storage and retrieval process in an OSD system.
34. Explain the storage and retrieval process for block,file,and object access in a unified storage system.
35. When is unified storage a suitable option for a data centre? Justify your answer.
36. Explain server clustering technology used in a data centre.
37. What are the various business/technical considerations for implementing a backup solution, and how do these considerations impact the choice of backup solutions/implementation.
38. List and explain the considerations in using tape as the backup technology.
39. Describe the benefits of using a virtual tape library over a physical tape library.
40. Discuss the benefits and challenges of using cloud storage for archiving.
41. Describe the uses of a local replica in various business operations.
42. What are the consideration for implementing synchronous remote replication?
43. Explain the RPO that can be achieved with synchronous, asynchronous, and disk buffered remote replication.
44. What are the essential characteristics of cloud computing?
45. How does cloud computing bring in business agility?
46. Explain cloud Orchestration.
47. What are the costs that should be evaluated to determine the financial advantages of cloud?
48. Develop a checklist for auditing the security of a storage environment with SAN,NAS ans iSCSI implementation.
49. Explain various security concerns and measure in the virtualized and cloud environment.
50. Explain multi factor authentication security techniques.

## Lesson Plan

### Unit-I

1. Introduction data, types of data ,big data,information,storage.
2. Evolution of Storage Architecture.
3. Data Centre Infrastructure.
4. Virtualization and Cloud Computing.
5. Data Centre Environment: Application.
6. DBMS, Host, Connectivity, Storage
7. Disk Drive Components, Disk Drive Performance.
8. Host Access to Data, Direct-Attached Storage.
9. Storage Design Based on Application.
10. Disk Native Command Queuing, Introduction to Flash Drives.

### Unit-II

1. Data Protection.
2. RAID Implementation Methods.
3. Array Components.
4. Techniques, Levels.
5. Impact on Disk Performance.
6. RAID Comparison.
7. Hot Spares.
8. Intelligent Storage System: Components.
9. Storage Provisioning.
10. Types of intelligent storage systems.

**Unit-III**

1. Fibre Channel Storage Area Networks.
2. FC Overview.
3. SAN and its Evolution.
4. Components of FC SAN.
5. FC Connectivity, Ports.
6. FC Architecture, Fabric Services.
7. Login Types, Zoning, FC SANTopologies.
8. Virtualization in SAN.
9. IP SAN and FcoE:
10. iSCSI.
11. FCIP.
12. FCoE.

**Unit-IV**

1. Network-Attached Storage.
2. Benefits, Components.
3. NAS I/O Operation.
4. NAS Implementations,.
5. File Sharing Protocols, Factors Affecting NAS Performance.
6. File-Level Virtualization.
7. Object Based and Unified Storage: Object Based Storage Devices.
8. Content Addressed Storage.
9. CAS Use Cases.
10. Unified Storage.

**Unit-V**

1. Business Continuity: Information Availability.
2. Terminology, Planning Lifecycle.
3. Failure Analysis.
4. Impact Analysis, Solutions.
5. Cloud Computing Infrastructure.
6. Securing the Storage Infrastructure: Framework, Risk Triad, Domains.
7. Security Implementations in Storage Networking, Virtualized and Cloud Environments.
8. Managing the Storage Infrastructure: Monitoring, Management Activities.
9. Management Challenges.
10. Information Lifecycle Management, Storage Tiering.

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcome** | | **Program Outcome(ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Explain storage architectures, and the logical and physical components of storage infrastructure including storage subsystems. | **L2** | - | M | H | - | L | - | M | H | - | L | - | - | - |
| Describe RAID and intelligent storage systems | **L1** | - | H | M | H | H | - | - | - | M | - | M | - | H |
| Illustrate storage networking technologies such as FC SAN, IP SAN, and FcoE | **L4** | - | H | H | M | M | - | - | - | - | H | H | - | M |
| Discuss NAS, and object-based and unified storage | **L6** | - | M | H | M | M | - | - | M | H | H | L | - | H |
| Explain business continuity, storage security, storage monitoring and management activities | **L2** | - | L | M | L | M | - | - | - | L | M | - | - | - |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcome** | | **Program Outcome(ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Explain storage architectures, and the logical and physical components of storage infrastructure including storage subsystems. | **L2** | - | 2 | 3 | - | 1 | - | 2 | 3 | - | 1 | - | - | - |
| Describe RAID and intelligent storage systems | **L1** | - | 3 | 2 | 3 | 3 | - | - | - | 2 | - | 2 | - | 3 |
| Illustrate storage networking technologies such as FC SAN, IP SAN, and FcoE | **L4** | - | 3 | 3 | 2 | 2 | - | - | - | - | 3 | 3 | - | 2 |
| Discuss NAS, and object-based and unified storage | **L6** | - | 2 | 3 | 2 | 2 | - | - | 2 | 3 | 3 | 1 | - | 3 |
| Explain business continuity, storage security, storage monitoring and management activities | **L2** | - | 1 | 2 | 1 | 2 | - | - | - | 1 | 2 | - | - | - |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |

| Course Title: J2EE & J2ME | | | |
|---|---|---|---|
| **Course Code: P13IS662** | **Semester : VI** | **L - T - P -H: 4-0-0-4** | **Credits:4** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites :** Object Oriented Programming with Java

## Course Learning Objectives

**This course aims to**

1. Analyze the real world problems to solve using object-oriented approach.
2. Develop Java applications to solve real world problems.
3. Create applets to solve real world problems.
4. Implement Multi Threading and Event Handing concepts.
5. Design GUIs using Swings.
6. Implement JDBC Concept.
7. Develop client server applications using Servlets and JSPs.
8. Implement RMI Concept.
9. Implement Enterprise Java Beans Concept.
10. Develop Mobile applications using J2ME.

## Course Content

### Unit – I

**Applets, Multi Threaded Programming, Event Handling:** The Applet Class: Two types of Applets; Applet basics; Applet Architecture; An Applet skeleton; Simple Applet display methods; Requesting repainting; Using the Status Window; The HTML APPLET tag; Passing parameters to Applets; getDocumentbase() and getCodebase();

Multi Threaded Programming: What are threads? How to make the classes threadable; Extending threads; Implementing runnable; Synchronization; Changing state of the thread; Bounded buffer problems, read-write problem. Event Handling: Two event handling mechanisms; The delegation event model; Event classes; Sources of events; Event listener interfaces; Using the delegation event model; Adapter classes; Inner classes. **11 Hrs**

### Unit – II

**Swings, Java 2 Enterprise Edition Overview, Database Access:** Swings: The origins of Swing; Two key Swing features; Components and Containers; The Swing Packages; A simple Swing Application; Create a Swing Applet; Jlabel and ImageIcon; JTextField;The Swing Buttons; JTabbedpane; JScrollPane; JList; JComboBox; JTable.

Overview of J2EE and J2SE, The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; ResultSet; Transaction Processing; Metadata, Data types; Exceptions.

**11 Hrs**

### Unit – III

**Servlets, Jsp:** Background; The Life Cycle of a Servlet; Using Tomcat for Servlet Development; A simple Servlet; The Servlet API; The Javax.servlet Package; Reading Servlet Parameter; The Javax.servlet.http package; Handling HTTP Requests and Responses; Using Cookies; Session Tracking.

Java Server Pages (JSP): JSP, JSP Tags, Tomcat, Request String, User Sessions, Cookies, Session Objects. **10 Hrs**

### Unit – IV

**Rmi, Enterprise Java Beans:** Java Remote Method Invocation: Remote Method Invocation concept; Server side, Client side. Enterprise java Beans; Deployment Descriptors; Session Java Bean, Entity Java Bean; Message-Driven Bean; The JAR File. **10 Hrs**

**Unit – V**

**J2ME:** Introduction, CDC, CLDC, MIDP; Programming for CLDC, MIDlet model, Provisioning, MIDlet life-cycle, Creating new application, MIDlet event handling, GUI in MIDP, Low level GUI Components, Multimedia APIs; Communication in MIDP, Security Considerations in MIDP. **10 Hrs**

**Text Books:**

1. Java - The Complete Reference – Herbert Schildt, 7th Edition, Tata McGraw Hill, 2007.
2. J2EE - The Complete Reference – James Keogh, Tata McGraw Hill, 2007.
3. Mobile Computing, Technology, Applications and Service Creation - Dr. Ashok Talukder, Ms Roopa Yavagal, Mr. Hasan Ahmed, 2nd Edition, Tata McGraw Hill, 2010

**References:**

1. Introduction to JAVA Programming – Y. Daniel Liang, 6th Edition, Pearson Education, 2007.
2. The J2EE Tutorial – Stephanie Bodoff et al, 2nd Edition, Pearson Education.
3. Mobile and Wireless Design Essentials - Martyn Mallik, Wiley.

## Course Outcomes

**After learning all the units of the course, the student is able to**

1. Recognize the object oriented concepts & apply them to create java applications.
2. Demonstrate java applications with applets, threads and event handling concepts.
3. Develop GUI applications with the help of swings.
4. Develop client server applications with JDBC, Servelts, JSP, RMI and EJB concepts.
5. Develop program for CLDC, MIDP let model and security concerns.

## Topic Learning Objectives

**After learning all the topics of unit – I, the student is able to**

1. Explain the life cycle of an applet – L2.
2. Write a Java program using applet to solve real world problems – L3.
3. Expain the use of getDocumentbase() and getCodebase() method – L2.
4. Explain the need of a Thread – L2.
5. Write a Java program using threads to solve real world problems – L3.
6. Explain the significance of synchronization in Java – L2.
7. Explain the delegation event model with an example – L2.
8. List out and explain the different methods in WindowListener – L1, L2.
9. Explain the need of an Inner Class – L2.
10. Write a Java program to illustarte event handling – L3.

**After learning all the topics of unit – II, the student is able to**

1. Explain the need of Swings – L2.
2. List four types of buttons in Swings with their use – L1, L2.
3. Write a Java program using Swings to solve real world problems – L3.
4. Explain J2EE multitier architecture – L2.
5. Explain the various JDBC driver types – L2.
6. Write a Java program using JDBC to solve real world problems – L3.
7. Explain the three types of getConnection method – L2.
8. Explain the use of scrollable result set with an example – L2.
9. Explain the use Metadata – L2.
10. Design client server applications with JDBC and Exception handling concepts.

**After learning all the topics of unit – III, the student is able to**

1. Explain the life cycle of a Servlet – L2.

2. Implement an interactive Web application using HTML forms and Servlets – L3.
3. Implement session management for a Servlet application – L3.
4. Write a program to manage cookies to store client-specific information using Servlets – L3.
5. Describe how a JSP is translated into a servlet and processed at runtime – L2.
6. Explain the use of directives on JSPs and outline the principal directives – L2.
7. Implement simple JSPs that use Java code in declarations, expressions and scriptlets – L3.
8. List out and use the implicit objects available to scripting elements – L1, L2.
9. Implement an interactive Web application using HTML forms and JSP – L3.
10. Implement session management for a JSP application – L3.
11. Write a program to manage cookies to store client-specific information using JSP – L3.

**After learning all the topics of unit – IV, the student is able to**

1. Explain the need of RMI – L2.
2. With the code snippet, explain RMI at server side – L3, L2.
3. With the code snippet, explain RMI at client side – L3, L2.
4. Develop client-server application using RMI – L3.
5. Explain the role of Enterprise JavaBeans in enterprise-level systems development, and its relationship to other J2EE technologies such as JSP and servlets – L2.
6. Explain the role of the EJB container in mediating contact between the client and the bean, transaction control, authorization control, and the importance of object pooling – L2.
7. Explain the role of entity beans, their lifecycle and interactions with the container – L2.
8. Explain the role of session beans, their lifecycle and interactions with the container – L2.
9. Develop and test stateless and stateful session beans – L3.
10. Explain the need of a JAR file – L2.

**After learning all the topics of unit – V, the student is able to**

1. Explain the need of J2ME – L2.
2. Explain CDC, CLDC and MIDP – L2.
3. Explain the MIDlet model – L2.
4. Explain MIDlet life-cycle – L2.
5. Write a program to illustrate the MIDlet event handling concept – L3.
6. Write a program to illustrate the creation of GUI in MIDP – L3.
7. List out and explain Low level GUI Components – L1, L2.
8. Write a program to solve real world problems using Multimedia APIs – L3.
9. Write a program to illustrate the concept of communication in MIDP – L3.
10. Write a program to illustrate the security considerations in MIDP concept – L3.

## Review Questions

1 Explain the different stages in the life cycle of an applet.
2 Write a Java applet that sets the background color to cyan and foreground color to red and outputs a string message “A Simple Java Applet”.
3 Write a program using an applet which will print “key pressed” on the status window when you press the key, “key released” on the status window when you release the key and when you type the characters it should print “Hello” at co-ordinates (50, 50) on applet.
4 Write a program to create multiple threads in Java.
5 Discuss the significance of synchronization in Java.
6 Explain how, the thread priority will be assigned with an example.
7 Explain the role of ActionEvent class with an example.
8 Explain the delegation event model with an example.
9 List out and explain the different methods in WindowListener.

10 Explain the adapter classes, with examples.
11 Differentiate between applets and swings.
12 Explain the MVC architecture of swings.
13 Explain the components and containers in the swings.
14 Explain the following: i) JTextField class ii) JButton class.
15 Write a program to create four different types of buttons on JApplet.
16 Explain the J2EE architecture.
17 Discuss about the various JDBC driver types.
18 Describe the various steps of JDBC, with code snippets.
19 With proper syntax, explain three types of getConnection method.
20 Explain the use of scrollable result set with an example.
21 Explain the life cycle of a servlet.
22 Describe in detail, how web server tomcat is configured for development of servlet.
23 With code snippet, explain how session tracking is handled in java with servlets.
24 List and explain the core classes that are provided in javax.servlet package
25 List out the applications of servlets.
26 List out the differences between servlets and JSP.
27 Explain the different types of JSP tags with syntax.
28 Write a JSP program to create and read cookie named "userid" that stores the value PES007.
29 With code snippet, explain how session tracking is handled in java with JSP.
30 With code snippet, explain how cookies are handled in java with JSP.
31 Explain the need of RMI.
32 Explain the working of RMI.
33 With the code snippet, explain RMI at server side.
34 With the code snippet, explain RMI at client side.
35 Design client server application using RMI concept.
36 List out and explain different kinds of enterprise beans.
37 List the deplyoment descriptor for EJB 1.1
38 Explain the functions of EJB transcation attributes.
39 Discuss the significance of session java bean.
40 Explain the need of a JAR file.
41 Explain the following: i) CDC ii) CLDC.
42 Define MIDP.
43 Explain MIDlet model.
44 Discuss MIDlet life-cycle with an example.
45 Explain MIDlet event handling with an example.
46 Illustrate the creation of GUI in MIDP.
47 List out and explain low level GUI Components.
48 Explain Multimedia APIs.
49 Illustrate the communication in MIDP with an example.
50 Demonstrate the security considerations in MIDP with an example.

## Lesson Plan

### Unit I

1 **Applets, Multi Threaded Programming, Event Handling:** The Applet Class: Two types of Applets; Applet basics;.
2 Applet Architecture; An Applet skeleton; Simple Applet display methods
3 Requesting repainting; Using the Status Window; The HTML APPLET tag
4 Passing parameters to Applets; getDocumentbase() and getCodebase()
5 Multi Threaded Programming: What are threads? How to make the classes threadable;

Extending threads.
6 Implementing runnable; Synchronization; Changing state of the thread
7 Bounded buffer problems, read-write problem.
8 Event Handling: Two event handling mechanisms; The delegation event model
9 Event classes; Sources of events; Event listener interfaces
10 Using the delegation event model; Examples
11 Adapter classes; Inner classes.

**Unit II**

1 **Swings, Java 2 Enterprise Edition Overview, Database Access:** Swings: The origins of Swing; Two key Swing features; Components and Containers
2 The Swing Packages; A simple Swing Application
3 Create a Swing Applet; Jlabel and ImageIcon
4 JTextField;The Swing Buttons; JTabbedpane
5 JScrollPane; JList; JComboBox; JTable.
6 Overview of J2EE and J2SE, The Concept of JDBC
7 JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process
8 Database Connection; Associating the JDBC/ODBC Bridge with the Database
9 Statement Objects; ResultSet; Transaction Processing
10 Programming Examples
11 Metadata, Data types; Exceptions.

**Unit III**

1 **Servlets, JSP:** Background; The Life Cycle of a Servlet.
2 Using Tomcat for Servlet Development; A simple Servlet.
3 The Servlet API; The Javax.servlet Package.
4 Reading Servlet Parameter; The Javax.servlet.http package.
5 Handling HTTP Requests and Responses
6 Using Cookies; Session Tracking.
7 Java Server Pages (JSP): JSP, JSP Tags
8 Using Tomcat server, practical demonstration
9 Request String, User Sessions.
10 Cookies, Session Objects.

**Unit IV**

1 **Rmi, Enterprise Java Beans:** Java Remote Method Invocation: Introduction, uses of RMI
2 Remote Method Invocation concept with an example
3 Remote Method Invocation concept, Server side with code snippet
4 Remote Method Invocation concept, Client side with code snippet
5 Programming examples with practical demonstration
6 Enterprise java Beans; Deployment Descriptors
7 Session Java Bean
8 Entity Java Bean
9 Message-Driven Bean; The JAR File.
10 Programming examples with practical demonstration

**Unit V**

1 **J2ME:** Introduction, CDC
2 CLDC and MIDP
3 Programming for CLDC
4 MIDlet model, Provisioning
5 MIDlet life-cycle, Creating new application

6 MIDlet event handling, GUI in MIDP
7 Low level GUI Components, Multimedia APIs
8 Communication in MIDP
9 Security Considerations in MIDP.
10 Examples

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome(ABET/NBA-(3a-m))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Recognize the object oriented concepts & apply them to create java applications. | **L2 L3** | M | L | M | | M | L | | | M | | M | | L |
| Demonstrate java applications with applets, threads and event handling concepts. | **L2** | | M | M | | M | | | | H | | M | | L |
| Develop GUI applications with the help of swings. | **L3** | | M | | | L | M | | | M | | M | | L |
| Develop client server applications with JDBC, Servelts, JSP, RMI and EJB concepts. | **L3** | M | M | M | | M | M | | | H | | M | | M |
| Develop program for CLDC, MIDP let model and security concerns. | **L3** | M | M | M | | M | M | | | M | | M | | L |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcomes** | | **Program Outcome (ABET/NBA-(3a-m))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Recognize the object oriented concepts & apply them to create java applications. | **L2 L3** | 2 | 1 | 2 | | 2 | 1 | | | 2 | | 2 | | 1 |
| Demonstrate java applications with applets, threads and event handling concepts. | **L2** | | 2 | 2 | | 2 | | | | 3 | | 2 | | 1 |
| Develop GUI applications with the help of swings. | **L3** | | 2 | | | 1 | 2 | | | 2 | | 2 | | 1 |
| Develop client server applications with JDBC, Servelts, JSP, RMI and EJB concepts. | **L3** | 2 | 2 | 2 | | 2 | 2 | | | 3 | | 2 | | 2 |
| Develop program for CLDC, MIDP let model and security concerns. | **L3** | 2 | 2 | 2 | | 2 | 2 | | | 2 | | 2 | | 1 |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |

| Course Title: Artificial Intelligence and Agent Technology | | | |
|---|---|---|---|
| **Course Code: P13IS663** | **Semester : VI** | **L- T – P -H: 4 – 0 – 0-4** | **Credits:4** |
| **Contact period : Lecture: 52 Hrs, Exam:3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites:** Operating System and Networks

**Course learning objectives(CLOs)**

**This course aims to**

1. Apply the AI technique to a given concrete problem.
2. Understand the concept of intelligent agent.
3. Analyze the problem to determine where it falls with respect to different issues.
4. Explain the knowledge based agent.
5. Develop the non-trivial AI techniques in a relatively large system
6. Understand uncertainty and Problem solving techniques.
7. Understand various symbolic knowledge representation to specify domains and reasoningtasks of a situated software agent.
8. Understand different logical systems for inference over formal domain representations, andtrace how a particular inference algorithm works on a given problem specification.
9. Understand various learning techniques and agent technology.
10. Explain the inductive learning.

**Course Content**

**Unit-I**

**Artificial Intelligence:** The AI Problems, The Underlying assumption, AI Technique, The Level of the model, Criteria for success, some general references, One final word. **Problems, problem spaces, and search:** Defining, the problem as a state space search, Production systems, Problem characteristics, Production system characteristics, Issues in the design of search programs, Additional Problems. **Intelligent Agents:** Agents and Environments, The nature of environments, The structure of agents. **11 Hrs**

**Unit-II**

**Heuristic search techniques:** Generate-and-test, Hill climbing, Best-first search, Problem reduction, Constraint satisfaction, Mean-ends analysis. Knowledge representation issues: Representations and mappings, Approaches to knowledge representation, Issues in knowledge representation, The frame problem. **Using predicate logic:** Representing simple facts in logic, representing instance and ISA relationships, Computable functions and predicates, Resolution, Natural Deduction. **Logical Agents:** Knowledge–based agents, the Wumpus world, Logic, Propositional logic, Reasoning patterns in Propositional Logic, Effective propositional model checking, Agents based on propositional logic.

**11 Hrs**

**Unit-III**

**Symbolic Reasoning Under Uncertainty:** Introduction to nonmonotonic reasoning, Logic for nonmonotonic reasoning, Implementation Issues, Augmenting a problem-solver, Implementation: Depthfirst search, Implementation: Breadth-first search. **Statistical Reasoning:** Probability and bayes Theorem, Certainty factors and rule-based systems, Bayesian Networks, Dempster-Shafer Theory, Fuzzy logic. **Uncertainty:** Acting under uncertainty, Basic probability notation, Inference using full joint distributions, Independence, Bayes' rule and its use, The Wumpus world revisited**.** **10 Hrs**

**Unit-IV**

**Weak Slot-and-filter structures:** Semantic Nets, Frames. **Strong slot-and-filler structures:** Conceptual dependency, scripts, CYC. **Knowledge Representation Summary:** Syntactic-Semantic spectrum of Representation, logic and slot-and-filler structures, other representational techniques, summary of the role of knowledge. **Game Playing:** Minimax search procedure, Adding alpha beta cutoffs, Adding Refinements, Iterative deepening. **10 Hrs**

**Unit-V**

**Learning From Observations:** Forms of learning, inductive learning, Learning decision trees, Evaluating and choosing the best hypothesis, The theory of learning ,PAC, Regression and Classification with linear models, Nonparametric models, Support vector machines, Ensemble learning. **StatisticalLearning Methods:** Statistical learning, learning with complete data, learning with hidden variables: The EM algorithm. **10 Hrs**

**Text Book :**

1. Elaine Rich, Kevin Knight, Shivashanka B Nair: Artificial Intelligence, Tata McGraw Hill, 3rd edition, 2013.
2. Stuart Russel, Peter Norvig: Artificial Intelligence A Modern Approach, Pearson, 2nd edition, 2012.

**Reference Books :**

1. Nils J. Nilsson: "Principles of Artificial Intelligence", Elsevier.

## Course outcomes

**After learning all the units of the course, the student is able to**

1. Understand the concept of artificial intelligence and intelligent agent.
2. Analyze the problem to determine where it falls with respect to different issues.
3. Understand uncertainty and Problem solving techniques.
4. Understand different logical systems for inference over formal domain representations, and trace how a particular inference algorithm works on a given problem specification.
5. Explain various learning techniques and agent technology.

## Topic learning objectives

**After learning all the topics of unit-I, the student is able to**

1. Understand the concept of AI.-L2
2. Understand the concept of intelligent agent.-L2
3. Identify the problem as a state space search.-L4
4. Distinguish between breadth first search and depth first search.-L4
5. Describe therole of knowledge. -L1
6. Explain the level of the model.-L2
7. Distinguish between simple reflex agents and model based reflex agents.-L4
8. Explain the concept of rationality.-L6
9. Explain the issues in the design of search programs.-L2
10. Explain the structure of agents. -L2

**After learning all the topics of unit-II, the student is able to**

1. Describe the problem to determine where it falls with respect to different issues. -L1
2. Discuss the algorithm for Generate and Test.-L6
3. Explain the knowledge based agent. -L2
4. Compare Simple Hill Climbing and Steepest Ascent Hill Climbing. -L2
5. Distinguish between A* algorithm and AO* algorithm. -L4
6. Explain Means-Ends analysis.-L2
7. Design the typical Wumpus world. -L5
8. Identify the agents based on propositional logic. -L4
9. Compare the inference based agent and circuit based agent. -L2
10. Analyze the knowledge based agents. -L4

**After learning all the topics of unit-III, the student is able to**

1. Explain the logics for Nonmonotonic reasoning. -L2

2. Identify the steps in Depth-First search. -L4
3. Design the context lattice of Breadth-first search. -L5
4. Design the Bayesian network with an example. -L5
5. Describe Bayes' theorem. -L3
6. Explain the Dempster-Shafer Theory. -L2
7. Design the Wumpus world revisited. -L5
8. Write the Bayes' rule and its use. -L5
9. Compare the uncertainty and rational decisions. -L2
10. Explain the axioms of probability. -L2

**After learning all the topics of unit-IV, the student is able to**

1. Explain the CYC. -L2
2. Design the simplified frame system with an example.-L5
3. Explain the representation of knowledge as constraints. -L2
4. Compare logic and slot-and-filler structures. -L2
5. Discuss the global ontology.-L6
6. Explain the minimax search procedure. -L2
7. Summarize on the Semantic Network. -L5
8. Write the algorithm for Minimax-A-B. -L5
9. Demonstrate the Alpha cutoff. -L3
10. Explain the iterative deepening. -L2

**After learning all the topics of unit-V, the student is able to**

1. Construct thedecision tree with an example. -L5
2. Explain the inductive learning. -L2
3. Compare single layer and multilayer neural network.-L2
4. Describe the Knowledge in learning. -L4
5. Describe the Ensemble learning. -L1
6. List the components of learning. -L1
7. Define neural network. -L1
8. Differentiate between nearest neighbor models and kernel modles. -L3
9. Explain the statistical learning. -L2
10. Explain the SVMs. -L2

**Review questions**

1. Explain the concept of AI.
2. Explain the concept of intelligent agent.
3. Explain the spectrum from static to AI-based techniques.
4. Distinguish between breadth first search and depth first search.
5. Describe the role of knowledge.
6. Explain how the branch and bound technique could be used to find the shortest solution to a water jug problem.
7. Distinguish between simple reflex agents and model based reflex agents.
8. Explain the concept of rationality.
9. What is an intelligent agent? Describe basic kinds of agents program.
10. Explain the structure of agents.
11. Prepare a note describing the role of machine intelligence in game playing.
12. Describe the problem to determine where it falls with respect to different issues.
13. Discuss the algorithm for Generate and Test.
14. Explain the knowledge based agent.
15. Write the brief notes on the following.

i. N-Queen Problem
ii. Hill climbing search.

16. Describe A* search technique. Prove that A* is complete and optimal.
17. Compare Simple Hill Climbing and Steepest Ascent Hill Climbing.
18. Distinguish between A* algorithm and AO* algorithm.
19. Design the typical Wumpus world.
20. Explain the knowledge based agents.
21. Describe the process of natural deduction for investigating the validity of an argument' Explain your answer by choosing a suitable example'
22. Describe Bayesian network technique of knowledge representation. How does it useful in representing uncertainty knowledge ?
23. What do you mean by logics for nonmonotonic reasoning? Explain.
24. What is probability and Baye's theorem? Explain.
25. Design the context lattice of Breadth-first search.
26. Explain the Dempster-Shafer Theory.
27. Design the Wumpus world revisited.
28. Compare the uncertainty and rational decisions.
29. Explain the axioms of probability.
30. Represent following sentences using symbolic logic:
    i. A drunker is enemy of himself.
    ii. Father of john loves to mother of Merry.
    iii. All students like a good teacher.
31. Explain the CYC.
32. Design the simplified frame system with an example.
33. Explain the representation of knowledge as constraints.
34. Construct semantic net representation for Pompeian (Marcus), Blacksmith (Marcus).
35. Construct a script for going to a movie from the viewpoint of the movie goer.
36. Discuss the global ontology.
37. Explain the minimax search procedure.
38. Write the algorithm for Minimax-A-B.
39. Demonstrate the Alpha cutoff.
40. Explain the iterative deepening.
41. Define decision tree.
42. Construct the decision tree with an example.
43. Explain the inductive learning.
44. Explain the multilayer neural network.
45. What is reinforcement learning? Differentiate between passive reinforcement learning and active reinforcement learning.
46. Describe the Ensemble learning.
47. Define neural network.
48. Differentiate between nearest neighbor models and kernel models.
49. Explain the statistical learning.
50. Explain the SVMs.

## Lesson plan

### Unit-I

1. Artificial Intelligence: The AI Problems, The Underlying assumption
2. AI Technique, The Level of the model
3. Criteria for success, some general references
4. One final word

5. Problems, problem spaces, and search: Defining, the problem as a state space search
6. Production systems, Problem characteristics
7. Production system characteristics, Issues in the design of search programs
8. Additional Problems
9. Intelligent Agents: Agents and Environments
10. The nature of environments
11. The structure of agents.

**Unit-II**

1. Heuristic search techniques: Generate-and-test, Hill climbing
2. Best-first search, Problem reduction, Constraint satisfaction
3. Mean-ends analysis.
4. Knowledge representation issues: Representations and mappings
5. Approaches to knowledge representation, Issues in knowledge representation
6. The frame problem
7. Using predicate logic: Representing simple facts in logic, representing instance and ISA relationships
8. Computable functions and predicates, Resolution, Natural Deduction
9. Logical Agents: Knowledge–based agents, the Wumpus world, Logic
10. Propositional logic, Reasoning patterns in Propositional Logic, Effective propositional model checking
11. Agents based on propositional logic.

**Unit-III**

1. Symbolic Reasoning Under Uncertainty: Introduction to nonmonotonic reasoning
2. Logic for nonmonotonic reasoning, Implementation Issues
3. Augmenting a problem-solver, Implementation: Depth first search
4. Implementation: Breadth-first search.
5. Statistical Reasoning: Probability and bayes Theorem
6. Certainty factors and rule-based systems, Bayesian Networks
7. Dempster-Shafer Theory, Fuzzy logic.
8. Uncertainty: Acting under uncertainty, Basic probability notation
9. Inference using full joint distributions, Independence
10. Bayes' rule and its use, The Wumpus world revisited.

**Unit-IV**

1. Weak Slot-and-filter structures: Semantic Nets
2. Frames
3. Strong slot-and-filler structures: Conceptual dependency
4. Scripts, CYC
5. Knowledge Representation Summary: Syntactic-Semantic spectrum of Representation
6. Logic and slot-and-filler structures
7. Other representational techniques, Summary of the role of knowledge
8. Game Playing: Minimax search procedure
9. Adding alpha beta cutoffs, Adding Refinements
10. Iterative deepening.

**Unit-V**

1. Learning From Observations: Forms of learning, inductive learning
2. Learning decision trees, Evaluating and choosing the best hypothesis
3. The theory of learning, PAC
4. Regression and Classification with linear models
5. Nonparametric models, Support vector machines
6. Ensemble learning

7. Statistical Learning Methods: Statistical learning
8. Learning with complete data
9. Learning with hidden variables: The EM algorithm
10. EM algorithm – continued.

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Understand the concept of artificial intelligence and intelligent agent. | **L2** | H | M | M | | M | | | | L | M | M | M | |
| Analyze the problem to determine where it falls with respect to different issues. | **L4** | M | | L | | H | | | | | L | M | | L |
| Understand uncertainty and Problem solving techniques. | **L2** | | M | L | | M | | | L | | L | M | L | |
| Understand different logical systems for inference over formal domain representations, and trace how a particular inference algorithm works on a given problem specification. | **L4** | H | H | M | M | M | | | L | | L | | L | |
| Explain various learning techniques and agent technology. | **L2** | M | M | M | | L | L | | | | | M | L | L |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |

| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Understand the concept of artificial intelligence and intelligent agent. | **L2** | 3 | 2 | 2 | | 2 | | | | 1 | 2 | 2 | 2 | |
| Analyze the problem to determine where it falls with respect to different issues. | **L4** | 2 | | 1 | | 3 | | | | | 1 | 2 | | 1 |
| Understand uncertainty and Problem solving techniques. | **L2** | | 2 | 1 | | 2 | | | 1 | | 1 | 2 | 1 | |
| Understand different logical systems for inference over formal domain representations, and trace how a particular inference algorithm works on a given problem specification. | **L4** | 3 | 3 | 2 | 2 | 2 | | | 1 | | 1 | | 1 | |
| Explain various learning techniques and agent technology. | **L2** | 2 | 2 | 2 | | 1 | 1 | | | | | 2 | 1 | 1 |
| **1 – Low, 2 – Moderate and 3 – High** | | | | | | | | | | | | | | |

| Course Title: Multimedia Computing | | | |
|---|---|---|---|
| **Course Code: P13IS664** | **Semester : VI** | **L- T – P -H: 4 – 0 – 0- 4** | **Credits:4** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

## Course Learning Objectives (CLOs)

**This course aims to**

1. Define multimedia.
2. Describe the components of multimedia systems, the technologies used for multimedia system, and storage management of multimedia system.
3. Understand applications of multimedia systems.
4. Describes the use of sound, music and speech in multimedia.
5. Describe the computerized graphic and images and their respective properties.
6. Understand characteristic and various formats to represent video and animations.
7. Understand different compression methods used for audio and video signals.
8. Describe the basic optical technology as well as techniques that represent the core of CD-DA, CD-ROM and DVD.
9. Explain different type of currently available content analysis techniques.
10. Describe a large number of file and data formats.

## Course Content

### Unit I

**Introduction, Media and Data Streams, Audio Technology**

Introduction: Multimedia Element, Multimedia Applications, Multimedia Systems Architecture, Evolving Technologies for Multimedia Systems, Defining objects for Multimedia systems, Multimedia data Interface standards, the need for data compression, Multimedia Databases.

Media: Perception media, Represent media, Presentation media, Storage media, Characterizing Continuous media data streams.

Sound: Frequency, Amplitude, Sound Perception and Psychoacoustics, Audio representation on Computers, Three dimensional sound projection, Music and MIDI standards, Speech signals, Speech Output, Speech Input, Speech Transmission. **10 Hrs**

### Unit II

**Graphics and Images, Video Technology, Computer-Based Animation**

Graphics and Images**:** Capturing Graphics and Images, Computer Assisted Graphics and Image Processing, Reconstructing Images, Graphics and Image output Options.

Basics, Television System, Digitalization of video Signals, Digital Television.

Computer-Based Animation: Basic Concepts, Specification of Animations, Methods of Controlling Animation, Virtual Reality Modeling Language. **11 Hrs**

### Unit III

**Data Compression**

Storage Space, Coding Requirements, Source, Entropy and Hybrid Coding, Basic Compression Techniques, JPEG: Image Preparation, Lossy Sequential DCT-based Mode, Expanded Lossy DCT-base Mode, Lossless Mode, Hierarchical Mode. H.261(Px64) and H.263: Image Preparation, Coding Algorithm, Data Stream, H.263+ and H.263L, MPEG: Video Encoding, Audio Encoding, Data Stream, MPEG-2,MPEG-4, MPEG-7,Fractal Compression. **10 Hrs**

### Unit IV

**Optical Storage Media**

History of Optical Storage, Basic Technology, Video Discs and Other WORMs, Compact Disc Digital Audio, Compact Disc Read Only Memory, CD-ROM Extended Architecture, Further CD-ROM Based

Developments, Compact Disc Recordable, Compact Disc Magneto-Optical, Compact Disc Read/Write, Digital Versatile Disc. **10 Hrs**

**Unit V**

**Content Analysis, Data and File Format standards**

Content Analysis: Simple Vs. Complex Features, Analysis of Individual Images, Analysis of Image Sequences, Audio Analysis, Applications.

Data and File Format standards: Rich Text Format, TIFF File Format, Resource Interchange File Format [RIFF], MIDI File Format. TWAIN. **11 Hrs**

**Text Books:**

1. Multimedia Fundamentals: Vol 1- Media Coding and Content Processing – Ralf Steinmetz, Klara Narstedt, 2nd Edition, Pearson Education / PHI, 2011.
2. Multimedia Systems Design - Prabhat K. Andleigh, Kiran Thakrar, PHI, 2013.

**Reference Books:**

1. Multimedia Communication: Applications, Networks, Protocols and Standards-Fred Halsal, Pearson, 2013
2. Introduction to Data Compression- Khalid Sayood, Fourth Edition, Elsevier Inc, 2013

**Course Outcome**

**After learning all the units of the course, the student is able to**

1. Define multimedia and describe the basic requirements of multimedia system.
2. Understand few properties, formats and standards to represent graphics and images, video and animations.
3. Describe different compression techniques used for audio and video.
4. Explain various optical storage media technology.
5. Understand content analysis techniques and various data and file formats standards.

**Topic learning objectives**

**After learning all the topics of unit-I, the student is able to**

1. Define multimedia. -L1
2. Define the basics of the various data elements and application. -L1
3. Explain the applications of multimedia. -L2
4. Design of multimedia system architecture. -L3
5. Define Objects of multimedia system. -L1
6. Explain different key attributes of multimedia and characteristics of data streams. -L2
7. Describe the storage management of multimedia system. -L2
8. Describe how audio material is represented in computers. -L1
9. Explain the formats used in audio technology. -L2
10. Explain speech signal with their input, output and transmission. -L2

**After learning all the topics of unit-II, the student is able to**

1. Describe computerized graphics and images and their respective properties. -L1
2. Explain how images can be acquired, manipulated and output on computers. -L2
3. Explain how the graphics and images represented and processed on computer. -L3
4. Describe some important formats of graphics and images. -L2
5. Explain methods used to retransform two-dimensional images into three dimensional spaces. -L2
6. Describe how to output graphics and images on output devices. -L2
7. Describe video technology and video standards with respect to properties of human perception. -L2

8. Explain the basic concepts of computer based animations. -L2
9. Describe the specification of animations. -L2
10. Explain the method of controlling animation. -L2

**After learning all the topics of unit-III, the student is able to**

1. Understand the storage space required and coding requirements of text, audio and video. -L1
2. Distinguish the compression techniques into three types of coding. -L3
3. Describe the major steps of data compression. -L2
4. Explain few basic compression techniques used in multimedia systems. -L2
5. Describe the JPEG compression technique for still images. -L2
6. Describe the H.263 (H.261 px64) compression technique for low resolution video sequences. -L2
7. Describe the MPEG compression technique for audio and video. -L2
8. Explain the MPEG-2 compression technique for audio and video. -L2
9. Explain the MPEG-4 compression standard. -L2
10. Describe the fractal compression techniques for fractal images and videos. -L2

**After learning all the topics of unit-IV, the student is able to**

1. Describe the history of Optical Storage. -L1
2. Describe the basic technology used in optical storage media. -L2
3. Explain briefly Video discs and other WORM's. -L2
4. Explain the basic technology, Eight-to-Fourteen modulation in CD-DA. -L2
5. Describe how the errors are handled in CD-DA. -L2
6. Explain CD-ROM and CD-ROM/XA based on CD-DA. -L2
7. Explain Compact Disc Recordable. -L2
8. Explain Compact Disc Magneto-optical and Compact Disc read/write. -L2
9. Explain Digital Versatile Disc (DVD). -L2
10. Compare CD and DVD. -L2

**After learning all the topics of unit-V, the student is able to**

1. Distinguish b/n simple and complex indicator features for content analysis. -L2
2. Understand techniques to analyse individual images and image sequences. -L3
3. Explain how audio analysis is performed using syntactic and semantic indicators. -L2
4. Understand applications of content analysis. -L2
5. Explain Rich Text Format. -L2
6. Describe Tagged Image File Format (TIFF). -L2
7. Explain Resource Interchange File Format (RIFF).- L2
8. Explain MIDI file format. -L2
9. Describe TWAIN specification objective. -L2
10. Describe Twain architecture. -L3

**Review Questions**

1. List any four benefits of multimedia system.
2. Describe the data elements for multimedia system.
3. Explain the multimedia workstation architecture with neat diagram.
4. Explain the issues related to multimedia storage and retrieval.
5. In detail, explain MIDI standards.
6. Explain the three characteristics for continuous media data stream that occur in multimedia systems.
7. Explain the following terms i) sampling rate ii) Quantization
8. With neat diagram, explain the principle of speech recognition.
9. Explain the need for data compression.
10. Explain how full motion video can be used in multimedia applications.

11. Describe the GIF and TIFF image file format.
12. Explain the concept of image segmentation.
13. Explain graphics and image output options.
14. Explain the coding methods used for digitization of video signal.
15. Define the following parameters w.r.t display monitor
    i) Resolution ii) Aspect ratio.
16. Describe various formats to represent video signals.
17. Write a note on VRML with example.
18. The VGA format can present 640x480 pixels with 256 simultaneous colors. The monitor is controlled via an analog RGB output. What is the storage capacity per frame required?
19. List and explain any four techniques for controlling animation.
20. Explain the two approaches for transmission of animations.
21. An uncompressed stereo audio signal of CD quality is sampled at 44.1 kHz and quantized using 16 bits. Calculate the data rate and space required to store **2**min audio clip.
22. The letters A,B,C,D and E are to be encoded and have frequency as follows A= 24, B=12, C=10, D=8, E=8 Use Huffman coding to derive a code for each letter.
23. Write a note on
    i) Run length coding ii) Vector quantization iii) Predictive or relative coding
23. With a neat diagram, explain the steps of lossy sequential DCT-mode of JPEG image compression.
24. Explain lossless and hierarchical JPEG modes.
25. Explain the two methods of coding in H.261.
26. Explain how the four types of images are coded in MPEG.
27. Explain the MPEG-4 system layer model.
28. Write a note on MPEG-7 and Fractal compression.
29. Explain the different layers of video stream in MPEG.
30. Explain hoe CD-DA works.
31. With neat diagram, explain the working principle and the process of error handling in CD-DA.
32. Describe the component of frames and areas in CD-DA.
33. Explain the mode-1 and mode-2 specification of CD-ROM.
34. In detail, explain Compact Disc Interactive.
35. Explain compact disc magneto-optical.
36. Explain the eight-to-fourteen modulation and process of error handling in CD-DA.
37. In detail, explain CD-R
38. Explain the 6 layers of DVD- video decoder to transfer MPEG data.
39. Describe the structure of DVD sector.
40. With a neat diagram, explain how the text recognition takes place in OCR systems.
41. Explain block and pixel oriented motion vector in image sequence analysis.
42. Explain the analysis of individual images in content analysis.
43. Explain the analysis of digital audio in content analysis.
44. Explain cut detection based on edge extraction.
45. Explain the different methods of controlling animation.
46. Describe the structure of TIFF and their tags.
47. Explain the key format information in RTF document files.
48. What are the TWAIN specification objectives?
49. Describe TWAIN architecture with neat diagram.

## Lesson plan

### Unit I

1. Multimedia Element, Multimedia Applications.
2. Multimedia Systems Architecture, Evolving Technologies for Multimedia Systems.

3. Defining objects for Multimedia systems, Multimedia data Interface standards.
4. The need for data compression, Multimedia Databases.
5. Perception media, Represent media, Presentation media, Storage media.
6. Characterizing Continuous media data streams.
7. Frequency, Amplitude, Sound Perception and Psychoacoustics.
8. Audio representation on Computers, Three dimensional sound projection.
9. Music and MIDI standards, Speech signals.
10. Speech Output, Speech Input, Speech Transmission.

**Unit II**

1. Introduction, Capturing Graphics and Images.
2. Capturing Graphics and Images.
3. Capturing Graphics and Images.
4. Computer Assisted Graphics and Image Processing.
5. Computer Assisted Graphics and Image Processing.
6. Reconstructing Images, Graphics and Image output Options.
7. Video Technology: Basics.
8. Video Technology: Basics.
9. Television System, Digitalization of video Signals, Digital Television.
10. **Computer-Based Animation:** Basic Concepts, Specification of Animations.
11. Methods of Controlling Animation, Virtual Reality Modeling Language

**Unit III**

1. Storage Space, Coding Requirements.
2. Source, Entropy and Hybrid Coding, Basic Compression Techniques.
3. Basic Compression Techniques.
4. JPEG: Image Preparation.
5. Lossy Sequential DCT-based Mode.
6. Expanded Lossy DCT-base Mode, Lossless Mode, Hierarchical Mode.
7. H.261(Px64) and H.263: Image Preparation, Coding Algorithm, Data Stream, H.263+ and H.263L
8. MPEG: Video Encoding, Audio Encoding, Data Stream
9. MPEG-2, MPEG-4
10. MPEG-4, MPEG-7,Fractal Compression

**Unit IV**

1. History of Optical Storage, Basic Technology.
2. Video Discs and Other WORMs, Compact Disc Digital Audio.
3. Compact Disc Digital Audio.
4. Compact Disc Read Only Memory.
5. CD-ROM Extended Architecture.
6. Further CD-ROM Based Developments.
7. Compact Disc Recordable.
8. Compact Disc Magneto-Optical.
9. Compact Disc Read/Write.
10. Digital Versatile Disc.

**Unit V**

1. Simple Vs. Complex Features.
2. Analysis of Individual Images.
3. Analysis of Image Sequences.
4. Analysis of Image Sequences.
5. Audio Analysis.
6. Applications.

7. Rich Text Format.
8. TIFF File Format.
9. Resource Interchange File Format [RIFF].
10. MIDI File Format.
11. TWAIN

| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcome (CO)** | | **Program Outcome(ABET/NBA-(3a-m))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Define multimedia and describe the basic requirements of multimedia system. | **L1 L2** | | L | | | | M | H | L | M | | M | | |
| Understand few properties, formats and standards to represent graphics and images, video and animations. | **L2** | | M | L | | | M | M | L | M | | M | | |
| Describe different compression techniques used for audio and video. | **L2** | | M | M | | M | M | H | L | M | | M | L | L |
| Explain various optical storage media technology. | **L2** | | | | M | | M | M | | M | | | | |
| Understand content analysis techniques and various data and file formats standards. | **L2 L3** | | M | | M | M | M | M | | M | | M | M | L |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome(ABET/NBA-(3a-m))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| Define multimedia and describe the basic requirements of multimedia system. | **L1 L2** | | 1 | | | | 2 | 3 | 1 | 2 | | 2 | | |
| Understand few properties, formats and standards to represent graphics and images, video and animations. | **L2** | | 2 | 1 | | | 2 | 2 | 1 | 2 | | 2 | | |
| Describe different compression techniques used for audio and video. | **L2** | | 2 | 2 | | 2 | 2 | 3 | 1 | 2 | | 2 | 1 | 1 |
| Explain various optical storage media technology. | **L2** | | | | 2 | | 2 | 2 | | 2 | | | | |
| Understand content analysis techniques and various data and file formats standards. | **L2 L3** | | 2 | | 2 | 2 | 2 | 2 | | 2 | | 2 | 2 | 1 |
| **1- Low, 2- Moderate, 3-High** | | | | | | | | | | | | | | |

| Course Title: Principles of Compiler Design | | | |
|---|---|---|---|
| **Course Code: P13IS665** | **Semester : VI** | **L- T – P-H : 4 – 0 – 0- 4** | **Credits:4** |
| **Contact period : Lecture: 52 Hrs, Exam: 3 Hrs** | | **Weightage: CIE: 50%; SEE: 50%** | |

**Prerequisites:** The student should have undergone the course on System Software and any one programming language**.**

**Course learning objectives**

**This course aims to**

1. Describe the various phases of a compiler, software tools to build compiler and know the technique used to implement lexical analyzer.
2. Apply an algorithm for top down parsing.
3. Apply an algorithm for bottom-up parsing.
4. Create a syntax-directed definition and explain the role of a semantic analyzer and type checking.
5. Apply the transformation of abstract syntax tree to intermediate code.
6. Analyze various issues in allocation strategies and design of code generator.
7. Implement the phases for set of instructions.
8. Apply the knowledge of LEX tool & YACC tool to develop a scanner & parser.
9. Design & conduct experiments for Intermediate Code Generation in compiler.
10. Acquire the knowledge of modern compiler & its features.
11. Learn & Use the new tools and technologies used for designing a compiler

**Course contents**

**Unit– I**

Language processors; The structure of a Compiler; Language processors; The evolution of programming languages; The science of building a Compiler; Applications of compiler technology; Programming language basics. Lexical analysis: The Role of Lexical Analyzer;
Input Buffering; Specifications of Tokens; Recognition of Tokens. **10 Hrs**

**Unit – II**

**Syntax Analysis – 1:** Introduction; Context-free Grammars; Writing a Grammar. Recursive decent parsing, FIRST set, FOLLOW set, LL(1) Grammar, Non-recursive predictive parsing, Error recovery in predictive parsing **10 Hrs**

**Unit – IV**

**Syntax Analysis – 2:** Reduction, Handle pruning, Shift reduce parsing, Conflicts during shift reduce parsing, Introduction to LR Parsing: Simple LR; LR parsing algorithm, Constructing SLR parsing tables, Constructing SLR parsing tables, Using ambiguous grammars; Parser Generators. **10 Hrs**

**Unit – V**

**Syntax-Directed Translation and Intermediate Code Generation** Syntax-directed definitions; Evaluation orders for SDDs; Applications of syntax-directed translation; Syntax-directed translation schemes.

**Intermediate Code Generation:** Variants of syntax trees; Three-address code; Translation of expressions; Control flow; Back patching; Switch statements; Procedure calls. **11 Hrs**

**Unit - V**

**Run-Time Environments and Code Generation** Storage Organization; Stack allocation of space; Access to non-local data on the stack; Heap management; Introduction to garbage collection.

**Code Generation:** Issues in the design of Code Generator; The Target Language; Addresses in the target code; Basic blocks and Flow graphs; Optimization of basic blocks; A Simple Code Generator **11 Hrs**

**Text Books:**

1. Alfred V Aho, Monica S.Lam, Ravi Sethi, Jeffrey D Ullman: Compilers- Principles, Techniques and Tools, 2nd Edition, Pearson Education, 2011.

**Reference Books:**

1. Charles N. Fischer, Richard J. leBlanc, Jr.: Crafting a Compiler with C, Pearson Education, 2009.
2. Andrew W Apple: Modern Compiler Implementation in C, Cambridge University Press, 2009.
3. Kenneth C Louden: Compiler Construction Principles & Practice, Cengage Learning, 2011.

## Course outcomes

**After learning all the units of the course, the student is able to**

1. Describe the various phases of a compiler, software tools to build compiler and know the technique used to implement lexical analyzer.
2. Apply an algorithm for top down parsing.
3. Apply an algorithm for bottom-up parsing .
4. Create a syntax-directed definition and apply the transformation of abstract syntax tree to intermediate code.
5. Analyze various issues in allocation strategies and design of code generator.

## Topic learning objectives

**After learning all the topics of unit-I, the student is able to**

1. Define the basic concepts of compiler. -L1
2. Describe the structure of a language processor.- L6
3. recognize the applications of compiler in programming language. -L2
4. Analyze the concept of compiler design and implementation.- L4
5. Discuss role of lexical analyzer.- L6
6. Differentiate between the buffer pairs and sentinels.- L4
7. Define token and understand the token specification.-L1
8. Understand and recognize tokens in lexical analysis. - L2

**After learning all the topics of unit-II, the student is able to**

1. Understand the role of the parser. -L2
2. Discuss the different error recovery strategies . -L6
3. Define CFG; understand how CFG is constructed use different notational conventions.-L1,L2
4. Discuss the ambiguity problems in context free grammars. -L6
5. Differentiate between lexical analysis and syntactic analysis. -L4
6. Analyze and understand ambiguity elimination. -L4
7. Analyze and understand Left Recursion problem. -L4
8. Discuss the recursive decent parsing .-L6
9. Understand the FIRST and FOLLOW set computation.-L2
10. Describe the LL(1) grammars.-L6

**After learning all the topics of unit-III, the student is able to**

1. Understand reductions in bottom-up parsing. -L2
2. Analyze handle pruning. -L4
3. Analyze and implement shift reduce parsing. -L4
4. Understand the shift reduce conflicts. -L2
5. Understand why LR parser. -L2
6. Discuss the LR parsing algorithm. -L6
7. Construct LALR parsing tables. -L3
8. Understand the error recovery in LR parsing. -L2

9. Discuss the parser generator using YACC.- L6

**After learning all the topics of unit-IV, the student is able to**

1. Differentiate between inherited attribute and synthesized attributes.-L4
2. Evaluate an SDD at the node of a parse tree.-L6
3. Discuss S- attributed definition.-L6
4. Discuss S- attributed definition.-L6
5. Understand the applications of SDD.-L2
6. Analyze the directed acyclic graphs for expression.-L4
7. Discuss the different three-address code types.-L6
8. Define type and declaration.-L1
9. Understand and implement code using switch statements.-L2, L5
10. Discuss the backpatching in intermediate code generation.-L6

**After learning all the topics of unit-V, the student is able to**

1. Differentiate static storage allocation and dynamic storage allocation.-L4
2. Discuss the stack allocation of space in run time .-L6
3. Define Heap.-L1
4. Analyze different Heap management factors.-L4
5. Discuss the design goals for garbage collectors.-L6
6. Discuss the different issues in the design of a code generator.-L6
7. Analyze the DAG representation of Basic blocks.-L4
8. Illustrate the flow graph representation.-L6
9. Understand the uses of Algebraic Identities.-L2
10. Apply reassembling basic blocks for optimization.-L3

**Review Questions**

1. Describe different phases of a compiler with the help of a neat diagram.
2. What are syntax trees? How these can be constructed?
3. Differentiate between phase and pass of a compiler.
4. Briefly explain the term “Compiler writing tools”.
5. What do you mean by cross compilation? Briefly explain.
6. What are assembler, compiler and interpreter? How these are related to each other?
7. Discuss the lexical analysis phase of the compiler with the help of an example.
8. What do you mean by a translator? Explain any three categories of translator.
9. Discuss any two problems which are encountered during code generation.
10. What are the characteristics of a good translator?
11. Describe algorithm to convert NDFA into DFA.
12. Construct a finite automaton accepting the set of all strings of zeros and ones; with at most one pair of consecutive zeros and at most one pair of consecutive ones.
13. What is the importance of lexical analyzer in a compiler?
14. What sets to strings do the following R.E$^2$s describe.
    a. {a b} {b/a}
    b. digit {digit}/ ϵ (0/2/4/6/8) where digit represents 0/1/2/ …… /9.
15. What is meant by input buffering? How is it useful in design of lexical analyzer?
16. What are Regular expressions? Discuss the procedure with example conversion of regular expression into finite automata and vice-versa. Also discuss how to minimize the number of states of a DFA.
17. Explain the syntax directed translation scheme for desk calculator. Also show the sequence of moves by parser for the input string 30/5+4$.

18. Explain the syntax directed translation scheme for desk calculator which performs +, * operations of simple data type. Also compute 23+5*45$.

19. Give a parse tree for the input string:
    i. i + i ↑ (i * i)#
20. What do you mean by a parse tree? How is it drawn? Explain with some example.
21. Define operator precedence grammar? How operator precedence relations are defined for a pair of terminals a and b.
22. What is a Top-Down parser? Consider the grammar S -> aSa/aa. By tracing through the steps of a top-down parser, which tries alternate aSa before aa, show that S succeeds on 2,4 or 8 a's but fails on 6 a's.
23. What do you understand by ambiguous grammar? How ambiguity can be removed? Explain with example.
24. What is intermediate code? What intermediate codes are used in compilers? Explain.
25. What do you understand by a handle? Explain the stack implementation of shift reduce parser with the help of example.
26. What is a context free grammar? Augmented grammar? What are the problems that may occur during parsing of these? Discuss the detail with an example.
27. Explain predictive parser. Also explain shift reduce parsing.
28. What are quadruples, triples and indirect triples? Give examples.
    a. Explain recursive-descent and predictive parsing.
29. Write the algorithm for construction of SLR parsing table. Write the sets of LR(0) items for the grammar
    i. E → E + T | T
    ii. T → T * F | F
    iii. F → (E)
    iv. T → id
30. Show that the following grammar
    i. S → Aa/bAc/dc/bda
    ii. A → d
    iii. Is LALR(1) but not SLR (1).
31. Convert the regular expression (a+b)*aba(a+b)* into the corresponding DFA.
32. Consider the grammar
    i. S → (L) | a
    ii. L → L, S | S
    1. What are the terminals, non-terminals and start symbol?
    2. Find parse tree for the following sentences: (a,a) (a, ((a,a),(a,a)))
33. Explain the function of LR parsers. Give algorithm for constructing SLR parsing table. Also construct the SLR table for grammar:
    i. Є → Є + T
    ii. Є → T
    iii. T → T * F
    iv. T → F
    v. F → (Є)
    vi. F → id
34. Write quadruples, triples and indirect triples for the expression:
    i. -(a + b) * (c + d) – (a + b + c)
35. Consider the grammar
    i. S → a | ^ | (T)
    ii. T → T, S | S

b. Find the right most derivation for (a,(a,a)).

36. Consider the grammar
    i. S → a | ^ | (T)
    ii. T → T, S | S
    b. Show the steps of a shift-reduce parser for the rightmost derivation.

37. Consider the grammar
    i. E → TE'
    ii. E'→ + T' | E
    iii. T → FT'
    iv. T'→ * FT' | E
    v. F→ (E) | id
    Compute FIRST and FOLLOW for each non-terminal of above grammar.
38. Explain in detail labeling algorithm used in code generation.
39. What are the contents of a symbol table? Give a brief description of each.
40. How symbol table space can be re-used. Give some example.
41. What are symbol tables? Explain their significance.
42. Write a detailed note on error detection and recovery.
43. Write about code optimization.
44. What is meant by register allocation? Why is considered to be important?
45. What do you mean by peephole optimization? What are the areas where it can be applied?
46. What is three address code? Write any five common kind of three address statements found in programs.
47. Write short notes on the following:
    A. Problems in code generation
    B. Register allocation and register assignment
48. Explain Backpatching, Bootstrapping and translation of assignment statement.
49. Write the algorithm for construction of SLR parsing table. Construct the SLR parsing table for the grammar.
    E → W + T | T
    T → T * F | F
    F → (E)/id
50. Explain Instruction variables and forms of object code.
51. Construct a DAG for the following basic block:
    D := B * C
    E := A + B
    B := B * C
    A := E – D
52. Compare the capabilities of context free grammars and regular expressions.

## Lesson Plan

### Unit-I

1. Language processors; The structure of a Compiler.
2. Language processors; The structure of a Compiler(continue)
3. The evolution of programming languages; The science of building a Compiler.
4. Applications of compiler technology.
5. Programming language basics.
6. Lexical analysis: The Role of Lexical Analyzer
7. Input Buffering
8. Specifications of Tokens

9. Recognition of Tokens.
10. Recognition of Tokens(continue)

**Unit –II**

1. Introduction
2. Context-free Grammars
3. Context free grammar (continue)
4. Writing a Grammar.
5. Recursive decent parsing
6. FIRST set
7. FOLLOW set
8. LL(1) Grammar
9. Non-recursive predictive parsing.
10. Error recovery in predictive parsing

**Unit – III**

1. Reduction.
2. Handle pruning.
3. Shift reduce parsing.
4. Conflicts during shift reduce parsing
5. Introduction to LR Parsing: Simple LR
6. LR parsing algorithm.
7. Constructing SLR parsing tables
8. Constructing SLR parsing tables
9. Using ambiguous grammars
10. Parser Generators.

**Unit-IV**

1. Syntax-directed definitions
2. Evaluation orders for SDDs
3. Applications of syntax-directed translation
4. Syntax-directed translation schemes.
5. Variants of syntax trees
6. Three-address code
7. Translation of expressions
8. Control flow
9. Back patching
10. Switch statements
11. Procedure calls.

**Unit-V**

1. Storage Organization
2. Stack allocation of space
3. Access to non-local data on the stack
4. Heap management
5. Introduction to garbage collection.
6. Code Generation: Issues in the design of Code Generator
7. The Target Language
8. Addresses in the target code
9. Basic blocks and Flow graphs
10. Optimization of basic blocks
11. A Simple Code Generator .

| **Course Articulation Matrix (CAM)** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Course Outcomes** | | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Describe** the various phases of a compiler, software tools to build compiler and **know** the technique used to implement lexical analyzer. | **L1 L2** | M | L | M | | | | | | L | | M | | M |
| **Apply** an algorithm for top down parsing. | **L2 L3 L4 L5** | H | H | H | | | | | | M | | M | H | M |
| **Apply** an algorithm for bottom up parsing. | **L2 L3 L4 L5** | H | H | H | | | | | | M | | M | H | M |
| **Create** a syntax-directed definition and **Explain** the role of a semantic analyzer and type checking. | **L2 L6** | L | M | M | | | | | | M | | M | | |
| **Apply** the transformation of abstract syntax tree to intermediate code. **Analyze** various issues in allocation strategies and design of code generator. | **L3 L4** | M | H | H | | | | | | M | | M | | M |
| **L- Low, M- Moderate, H-High** | | | | | | | | | | | | | | |
| **Course Assessment Matrix (CAM)** | | | | | | | | | | | | | | |
| **Course Outcomes** | | **Program Outcome (ABET/NBA-(3a-k))** | | | | | | | | | | | | |
| | | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **k** | **l** | **m** |
| **Describe** the various phases of a compiler, software tools to build compiler and **know** the technique used to implement lexical analyzer. | **L1 L2** | 2 | 1 | 2 | | | | | | 1 | | 2 | | 2 |
| **Apply** an algorithm for top down parsing. | **L2 L3 L4 L5** | 3 | 3 | 3 | | | | | | 2 | | 2 | 3 | 2 |
| **Apply** an algorithm for bottom up parsing. | **L2 L3 L4 L5** | 3 | 3 | 3 | | | | | | 2 | | 2 | 3 | 2 |
| **Create** a syntax-directed definition and **Explain** the role of a semantic analyzer and type checking. | **L2 L6** | 1 | 2 | 2 | | | | | | 2 | | 2 | | |
| **Apply** the transformation of abstract syntax tree to intermediate code. **Analyze** various issues in allocation strategies and design of code generator. | **L3 L4** | 2 | 3 | 3 | | | | | | 2 | | 2 | | 2 |
| **1 – Low, 2 – Moderate and 3 – High** | | | | | | | | | | | | | | |