<table>
<thead>
<tr>
<th>Module</th>
<th>Subject</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 1</td>
<td>Fourier Series: Periodic functions, Dirichlet’s condition, Fourier Series of periodic functions with period $2\pi$ and with arbitrary period $2c$. Fourier series of even and odd functions. Half range Fourier Series, practical harmonic analysis-Illustrative examples from engineering field.</td>
<td>10 Hours</td>
</tr>
<tr>
<td>Module 2</td>
<td>Fourier Transforms: Infinite Fourier transforms, Fourier sine and cosine transforms. Inverse Fourier transform. Z-transform: Difference equations, basic definition, z-transform-definition, Standard z-transforms, Damping rule, Shifting rule, Initial value and final value theorems (without proof) and problems, Inverse z-transform. Applications of z-transforms to solve difference equations.</td>
<td>10 Hours</td>
</tr>
<tr>
<td>Module 4</td>
<td>Finite differences: Forward and backward differences, Newton’s forward and backward interpolation formulae. Divided differences- Newton’s divided difference formula. Lagrange’s interpolation formula and inverse interpolation formula (all formulae without proof)-Problems. Numerical integration: Simpson’s $(1/3)^{th}$ and $(3/8)^{th}$ rules, Weddle’s rule (without proof ) – Problems.</td>
<td>10 Hours</td>
</tr>
<tr>
<td>Module 5</td>
<td>Vector integration: Line integrals-definition and problems, surface and volume integrals-definition, Green’s theorem in a plane, Stokes and Gauss-divergence theorem(without proof) and problems. Calculus of Variations: Variation of function and Functional, variational problems. Euler’s equation, Geodesics, hanging chain, problems.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

Course outcomes:
After Studying this course, students will be able to

- Know the use of periodic signals and Fourier series to analyze circuits and system communications.
- Explain the general linear system theory for continuous-time signals and digital signal processing using the Fourier Transform and z-transform.
- Employ appropriate numerical methods to solve algebraic and transcendental equations.
- Apply Green's Theorem, Divergence Theorem and Stokes' theorem in various applications in the field of electro-magnetic and gravitational fields and fluid flow problems.
- Determine the extremals of functionals and solve the simple problems of the calculus of variations.

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

## ANALOG AND DIGITAL ELECTRONICS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)

### SEMESTER - III

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>17CS32</th>
<th>IA Marks</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>04</td>
<td>Exam Marks</td>
<td>60</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>50</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

### Module -1

**Field Effect Transistors:** Junction Field Effect Transistors, MOSFETs, Differences between JFETs and MOSFETs, Biasing MOSFETs, FET Applications, CMOS Devices. Wave-Shaping Circuits: Integrated Circuit (IC) Multivibrators. **Introduction to Operational Amplifier:** Ideal v/s practical Opamp, Performance Parameters. **Operational Amplifier Application Circuits:** Peak Detector Circuit, Comparator, Active Filters, Non-Linear Amplifier, Relaxation Oscillator, Current-To-Voltage Converter, Voltage-To-Current Converter.


### Module -2

**The Basic Gates:** Review of Basic Logic gates, Positive and Negative Logic, Introduction to HDL. **Combinational Logic Circuits:** Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications, Don’t-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method, Hazards and Hazard covers, HDL Implementation Models.

**Text book 2:** Ch2: 2.4, 2.5. Ch3: 3.2 to 3.11.

### Module – 3

**Data-Processing Circuits:** Multiplexers, Demultiplexers, 1-of-16 Decoder, BCD to Decimal Decoders, Seven Segment Decoders, Encoders, Exclusive-OR Gates, Parity Generators and Checkers, Magnitude Comparator, Programmable Array Logic, Programmable Logic Arrays, HDL Implementation of Data Processing Circuits. Arithmetic Building Blocks, Arithmetic Logic Unit **Flip- Flops:** RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS FLIP-FLOP, Edge-triggered D FLIP-FLOPs, Edge-triggered JK FLIP-FLOPs.

**Text book 2:** Ch 4:- 4.1 to 4.9, 4.11, 4.12, 4.14.Ch6:-6.7, 6.10.Ch8:- 8.1 to 8.5.

### Module-4

**Flip- Flops:** FLIP-FLOP Timing, JK Master-slave FLIP-FLOP, Switch Contact Bounce Circuits, Various Representation of FLIP-FLOPs, HDL Implementation of FLIP-FLOP. **Registers:** Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers, Register implementation in HDL. **Counters:** Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus.

**Text book 2:** Ch 8: 8.6, 8.8, 8.9, 8.10, 8.13. Ch 9: 9.1 to 9.8. Ch 10: 10.1 to 10.4
| Module-5 |
|------------------|------------------|------------------|
| **Counters:** Decade Counters, Presettable Counters, Counter Design as a Synthesis problem, A Digital Clock, Counter Design using HDL. **D/A Conversion and A/D Conversion:** Variable, Resistor Networks, Binary Ladders, D/A Converters, D/A Accuracy and Resolution, A/D Converter-Simultaneous Conversion, A/D Converter-Counter Method, Continuous A/D Conversion, A/D Techniques, Dual-slope A/D Conversion, A/D Accuracy and Resolution. | 10 Hours |
| **Text book 2:** Ch 10: 10.5 to 10.9. Ch 12: 12.1 to 12.10 |

**Course outcomes:** After Studying this course, students will be able to

- Explain the operation of JFETs and MOSFETs, Operational Amplifier circuits and their application
- Explain Combinational Logic, Simplification Techniques using Karnaugh Maps, Quine McClusky technique.
- Demonstrate Operation of Decoders, Encoders, Multiplexers, Adders and Subtractors, working of Latches, Flip-Flops, Designing Registers, Counters, A/D and D/A Converters
- Design of Counters, Registers and A/D & D/A converters

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

### DATA STRUCTURES AND APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

**SEMESTER - III**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>17CS33</th>
<th>IA Marks</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>04</td>
<td>Exam Marks</td>
<td>60</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>50</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS - 04**

#### Module -1


- **Text 1:** Ch 1: 1.2, Ch2: 2.2 - 2.7
- **Text 2:** Ch 1: 1.1 - 1.4, Ch 3: 3.1-3.3,3.5,3.7, Ch 4: 4.1-4.9,4.14
- **Ref 3:** Ch 1: 1.4

#### Module -2

**Stacks and Queues**

**Stacks:** Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression, **Recursion** - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. **Queues:** Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues, A Mazing Problem. Multiple Stacks and Queues. Programming Examples.

- **Text 1:** Ch3: 3.1 - 3.7
- **Text 2:** Ch6: 6.1 - 6.3, 6.5, 6.7-6.10, 6.12, 6.13

#### Module – 3

**Linked Lists:** Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. Programming Examples

- **Text 1:** Ch4: 4.1 - 4.8 except 4.6
- **Text 2:** Ch5: 5.1 – 5.10

| Teaching Hours | 10 Hours |
### Module-4

**Text 1:** Ch5: 5.1 –5.5, 5.7  
**Text 2:** Ch7: 7.1 – 7.9  

<table>
<thead>
<tr>
<th>10 Hours</th>
</tr>
</thead>
</table>

### Module-5

**Graphs:** Definitions, Terminologies, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search.  
**Sorting and Searching:** Insertion Sort, Radix sort, Address Calculation Sort. **Hashing:** Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. **Files and Their Organization:** Data Hierarchy, File Attributes, Text Files and Binary Files, Basic File Operations, File Organizations and Indexing  
**Text 1:** Ch6: 6.1 –6.2, Ch 7:7.2, Ch 8:8.1-8.3  
**Text 2:** Ch8: 8.1 – 8.7, Ch 9:9.1-9.3,9.7,9.9  
**Reference 2:** Ch 16: 16.1 - 16.7  

<table>
<thead>
<tr>
<th>10 Hours</th>
</tr>
</thead>
</table>

### Course outcomes:
After studying this course, students will be able to:
- Explain different types of data structures, operations and algorithms
- Apply searching and sorting operations on files
- Make use of stack, Queue, Lists, Trees and Graphs in problem solving.
- Develop all data structures in a high-level language for problem solving.

### Question paper pattern:
The question paper will have ten questions.  
There will be 2 questions from each module.  
Each question will have questions covering all the topics under a module.  
The students will have to answer 5 full questions, selecting one full question from each module.

### Text Books:

### Reference Books:
### COMPUTER ORGANIZATION

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

**SEMESTER - III**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS34</td>
<td>40</td>
<td>60</td>
<td>03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>40</td>
<td>60</td>
<td>03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Number of Lecture Hours</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>40</td>
<td>60</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

<table>
<thead>
<tr>
<th>Module -1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input and Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Encoding of Machine Instructions</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module -2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input/Output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Direct Memory Access, Buses Interface Circuits, Standard I/O Interfaces – PCI Bus, SCSI Bus, USB.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory System: Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size, and Cost, Cache Memories – Mapping Functions, Replacement Algorithms, Performance Considerations, Virtual Memories, Secondary Storage.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module-4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic: Numbers, Arithmetic Operations and Characters, Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed Operand Multiplication, Fast Multiplication, Integer Division, Floating-point Numbers and Operations.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module-5</th>
<th>Teaching Hours</th>
</tr>
</thead>
</table>

**Course outcomes:** After studying this course, students will be able to:

- Explain the basic organization of a computer system.
- Demonstrate functioning of different sub systems, such as processor, Input/output, and memory.
- Illustrate hardwired control and micro programmed control, pipelining, embedded and other computing systems.
- Build simple arithmetic and logical units.
**Question paper pattern:**
The question paper will have ten questions. 
There will be 2 questions from each module. 
Each question will have questions covering all the topics under a module. 
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**
# UNIX AND SHELL PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2017 -2018)  

**SEMESTER – III**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>17CS35</th>
<th>IA Marks</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>03</td>
<td>Exam Marks</td>
<td>60</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

<table>
<thead>
<tr>
<th>Module</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module -1</strong></td>
<td>08 Hours</td>
</tr>
<tr>
<td>Introduction, Brief history. Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. The login prompt. General features of Unix commands/ command structure. Command arguments and options. Understanding of some basic commands such as echo, printf, ls, who, date, passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The man command knowing more about Unix commands and using Unix online manual pages. The man with keyword option and whatis. The more command and using it with other commands. Knowing the user terminal, displaying its characteristics and setting characteristics. Managing the non-uniform behaviour of terminals and keyboards. The root login. Becoming the super user: su command. The /etc/passwd and /etc/shadow files. Commands to add, modify and delete users.</td>
<td>08 Hours</td>
</tr>
<tr>
<td><strong>Module -2</strong></td>
<td>08 Hours</td>
</tr>
<tr>
<td><strong>Module –3</strong></td>
<td>08 Hours</td>
</tr>
</tbody>
</table>

**Topics from chapter 2, 3 and 15 of text book 1, chapter 1 from text book 2**

**Topics from chapters 4, 5 and 6 of text book 1**

**Topics from chapters 7, 8 and 13 of text book 1. Topics from chapter 2 and 9 ,10 of text book 2**
### Module-4


**Topics from chapter 11, 12, 14 of text book 1,chapter 17 from text book2**

### Module-5


**Topics from chapter 9 and 19 of text book 1. Topics from chapter 11 of reference book 1**

### Course outcomes:

After studying this course, students will be able to:

- Explain UNIX system and use different commands.
- Compile Shell scripts for certain functions on different subsystems.
- Demonstrate use of editors and Perl script writing

### Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

### Text Books:


### Reference Books:

# DISCRETE MATHEMATICAL STRUCTURES

As per Choice Based Credit System (CBCS) scheme  
(Effective from the academic year 2017-2018)  
SEMESTER – III

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>17CS36</th>
<th>CREDITS – 04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>04</td>
<td>50</td>
<td>04</td>
</tr>
<tr>
<td>Exam Marks</td>
<td>60</td>
<td>3</td>
<td>03</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>50</td>
<td>10</td>
<td>04</td>
</tr>
</tbody>
</table>

## Module -1

**Fundamentals of Logic:** Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. Fundamentals of Logic contd.: The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems,  
**Teaching Hours:** 10

## Module -2

**Teaching Hours:** 10

## Module – 3

**Relations and Functions:** Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions, The Pigeon-hole Principle, Function Composition and Inverse Functions, Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions.  
**Teaching Hours:** 10

## Module-4

**The Principle of Inclusion and Exclusion:** The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials. **Recurrence Relations:** First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients,  
**Teaching Hours:** 10

## Module-5

**Introduction to Graph Theory:** Definitions and Examples, Sub graphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits . **Trees:** Definitions, Properties, and Examples, Routed Trees, Trees and Sorting, Weighted Trees and Prefix Codes  
**Teaching Hours:** 10

## Course outcomes:

After studying this course, students will be able to:

- Make use of propositional and predicate logic in knowledge representation and truth verification.
- Demonstrate the application of discrete structures in different fields of computer science.
- Solve problems using recurrence relations and generating functions.
- Apply different mathematical proofs, techniques in proving theorems.
- Compare graphs, trees and their applications.
**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

<table>
<thead>
<tr>
<th><strong>Text Books:</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Reference Books:</strong></th>
</tr>
</thead>
</table>
ANALOG AND DIGITAL ELECTRONICS LABORATORY  
[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2017 -2018)  
SEMESTER - III

<table>
<thead>
<tr>
<th>Laboratory Code</th>
<th>IA Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CSL37</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>01I + 02P</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>03</td>
<td>02</td>
</tr>
</tbody>
</table>

CREDITS – 02

Descriptions (if any)

Any simulation package like MultiSim / P-spice /Equivalent software may be used. Faculty-in-charge should demonstrate and explain the required hardware components and their functional Block diagrams, timing diagrams etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-up on analog components; functional block diagram, Pin diagram (if any), waveforms and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-up on Logic design components, pin diagram (if any), Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

Note: These TWO Laboratory sessions are used to fill the gap between theory classes and practical sessions. Both sessions are to be evaluated for 40 marks as lab experiments.

Laboratory Experiments:

1. a) Design and construct a Schmitt trigger using Op-Amp for given UTP and LTP values and demonstrate its working.  
   b) Design and implement a Schmitt trigger using Op-Amp using a simulation package for two sets of UTP and LTP values and demonstrate its working.

2. a) Design and construct a rectangular waveform generator (Op-Amp relaxation oscillator) for given frequency and demonstrate its working.  
   b) Design and implement a rectangular waveform generator (Op-Amp relaxation oscillator) using a simulation package and demonstrate the change in frequency when all resistor values are doubled.

3. Design and implement an Astable multivibrator circuit using 555 timer for a given frequency and duty cycle.

NOTE: hardware and software results need to be compared


5. a) Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.  
   b) Design and develop the Verilog/VHDL code for an 8:1 multiplexer. Simulate and verify its working.

7. Design and verify the Truth Table of 3-bit Parity Generator and 4-bit Parity Checker using basic Logic Gates with an even parity bit.

8. a) Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table.
   b) Design and develop the Verilog / VHDL code for D Flip-Flop with positive-edge triggering. Simulate and verify it’s working.

9. a) Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
   b) Design and develop the Verilog / VHDL code for mod-8 up counter. Simulate and verify it’s working.

10. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate on 7-segment display (using IC-7447).

11. Generate a Ramp output waveform using DAC0800 (Inputs are given to DAC through IC74393 dual 4-bit binary counter).

**Study experiment**

12. To study 4-bit ALU using IC-74181.

**Course outcomes:**
On the completion of this laboratory course, the students will be able to:
- Demonstrate various Electronic Devices like Cathode ray Oscilloscope, Signal generators, Digital Trainer Kit, Multimeters and components like Resistors, Capacitors, Op amp and Integrated Circuit.
- Design and demonstrate various combinational logic circuits.
- Design and demonstrate various types of counters and Registers using Flip-flops
- Make use of simulation package to design circuits.
- Infer the working and implementation of ALU.

**Conduction of Practical Examination:**

1. All laboratory experiments (1 to 11 nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution:
   a) For questions having part a only- Procedure + Conduction + Viva: 15 + 70 + 15 =100 Marks
   b) For questions having part a and b
      Part a- Procedure + Conduction + Viva: 09 + 42 +09= 60 Marks
      Part b- Procedure + Conduction + Viva: 06 + 28 +06= 40 Marks
5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

DATA STRUCTURES LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)

SEMESTER - III

<table>
<thead>
<tr>
<th>Laboratory Code</th>
<th>17CSL38</th>
<th>IA Marks</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>01I + 02P</td>
<td>Exam Marks</td>
<td>60</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

CREDITS - 02

Descriptions (if any)

Implement all the experiments in C Language under Linux / Windows environment.

Laboratory Experiments:

1. Design, Develop and Implement a menu driven Program in C for the following **Array** operations
   a. Creating an Array of **N** Integer Elements
   b. Display of Array Elements with Suitable Headings
   c. Inserting an Element (**ELEM**) at a given valid Position (**POS**)  
   d. Deleting an Element at a given valid Position (**POS**) 
   e. Exit.
   Support the program with functions for each of the above operations.

2. Design, Develop and Implement a Program in C for the following **operation**son **Strings**
   a. Read a main String (**STR**), a Pattern String (**PAT**) and a Replace String (**REP**) 
   b. Perform Pattern Matching Operation: Find and Replace all occurrences of **PAT** in **STR** with **REP** if **PAT** exists in **STR**. Report suitable messages in case **PAT** does not exist in **STR**
   Support the program with functions for each of the above operations. Don't use Built-in functions.

3. Design, Develop and Implement a Program in C for the following **operation**s on **Stack** of Integers (Array Implementation of Stack with maximum size **MAX**) 
   a. **Push** an Element on to Stack 
   b. **Pop** an Element from Stack 
   c. Demonstrate how Stack can be used to check **Palindrome** 
   d. Demonstrate **Overflow** and **Underflow** situations on Stack 
   e. Display the status of Stack 
   f. Exit
   Support the program with appropriate functions for each of the above operations

4. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.

5. Design, Develop and Implement a Program in C for the following **Stack Application**s 
   a. Evaluation of **Suffix expression** with single digit operands and operators: +, -, *, /, %, ^
   b. Solving **Tower of Hanoi** problem with **n** disks
6. Design, Develop and Implement a menu driven Program in C for the following operations on **Circular QUEUE** of Characters (Array Implementation of Queue with maximum size MAX)
   a. Insert an Element on to Circular QUEUE
   b. Delete an Element from Circular QUEUE
   c. Demonstrate **Overflow** and **Underflow** situations on Circular QUEUE
   d. Display the status of Circular QUEUE
   e. Exit
   Support the program with appropriate functions for each of the above operations

7. Design, Develop and Implement a menu driven Program in C for the following operations on **Singly Linked List (SLL)** of Student Data with the fields: **USN, Name, Branch, Sem, PhNo**
   a. Create a SLL of N Students Data by using **front insertion**.
   b. Display the status of SLL and count the number of nodes in it
   c. Perform Insertion / Deletion at End of SLL
   d. Perform Insertion / Deletion at Front of SLL (**Demonstration of stack**)
   e. Exit

8. Design, Develop and Implement a menu driven Program in C for the following operations on **Doubly Linked List (DLL)** of Employee Data with the fields: **SSN, Name, Dept, Designation, Sal, PhNo**
   a. Create a DLL of N Employees Data by using **end insertion**.
   b. Display the status of DLL and count the number of nodes in it
   c. Perform Insertion and Deletion at End of DLL
   d. Perform Insertion and Deletion at Front of DLL
   e. Demonstrate how this DLL can be used as **Double Ended Queue**
   f. Exit

9. Design, Develop and Implement a Program in C for the following operations on **Singly Circular Linked List (SCLL)** with header nodes
   a. Represent and Evaluate a Polynomial \( P(x,y,z) = 6x^2y^2z-4yz^5+3x^3yz+2xy^5z-2xyz^3 \)
   b. Find the sum of two polynomials **POLY1**(x,y,z) and **POLY2**(x,y,z) and store the result in **POLYSUM(x,y,z)**
   Support the program with appropriate functions for each of the above operations

10. Design, Develop and Implement a menu driven Program in C for the following operations on **Binary Search Tree (BST)** of Integers
    a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
    b. Traverse the BST in Inorder, Preorder and Post Order
    c. Search the BST for a given element (KEY) and report the appropriate message
    e. Exit

11. Design, Develop and Implement a Program in C for the following operations on **Graph(G)** of Cities
    a. Create a Graph of N cities using Adjacency Matrix.
    b. Print all the nodes **reachable** from a given starting node in a digraph using DFS/BFS method
12. Given a File of \( N \) employee records with a set \( K \) of Keys (4-digit) which uniquely determine the records in file \( F \). Assume that file \( F \) is maintained in memory by a Hash Table (HT) of \( m \) memory locations with \( L \) as the set of memory addresses (2-digit) of locations in HT. Let the keys in \( K \) and addresses in \( L \) are Integers. Design and develop a Program in C that uses Hash function \( H: K \rightarrow L \) as \( H(K) = K \mod m \) (remainder method), and implement hashing technique to map a given key \( K \) to the address space \( L \). Resolve the collision (if any) using linear probing.

**Course outcomes:**
On the completion of this laboratory course, the students will be able to:

- Analyze and Compare various linear and non-linear data structures
- Demonstrate the working nature of different types of data structures and their applications
- Develop, analyze and evaluate the searching and sorting algorithms
- Choose the appropriate data structure for solving real world problems

**Conduction of Practical Examination:**
1. All laboratory experiments (TWELVE nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script
4. Marks distribution: Procedure + Conduction + Viva: 15 + 70 + 15 (100)
5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.
# ENGINEERING MATHEMATICS-IV

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)

## SEMESTER – IV

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17MAT41</td>
<td>40</td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

| CREDITS – 04 |

## Module 1

**Numerical Methods:** Numerical solution of ordinary differential equations of first order and first degree, Taylor’s series method, modified Euler’s method. Runge - Kutta method of fourth order, Milne’s and Adams-Bashforth predictor and corrector methods (No derivations of formulae-single step computation only).

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

## Module 2

**Numerical Methods:** Numerical solution of second order ordinary differential equations, Runge-Kutta method and Milne’s method. (No derivations of formulae-single step computation only).

**Special Functions:** Series solution of Bessel’s differential equation leading to $J_n(x)$-Bessel’s function of first kind. Basic properties and orthogonality. Series solution of Legendre’s differential equation leading to $P_n(x)$-Legendre polynomials. Rodrigue’s formula, problems

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

## Module 3

**Complex Variables:** Review of a function of a complex variable, limits, continuity, differentiability. Analytic functions-Cauchy-Riemann equations in cartesian and polar forms. Properties and construction of analytic functions. Complex line integrals-Cauchy’s theorem and Cauchy’s integral formula, Residue, poles, Cauchy’s Residue theorem (without proof) and problems.

**Transformations:** Conformal transformations-Discussion of transformations: $w = z^2$, $w = e^z$, $w = z + (1/z) (z \neq 0)$, Bilinear transformations-problems

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

## Module 4

**Probability Distributions:** Random variables (discrete and continuous), probability functions. Poisson distributions, geometric distribution, uniform distribution, exponential and normal distributions, Problems. **Joint probability distribution:** Joint Probability distribution for two variables, expectation, covariance, correlation coefficient.

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

## Module 5

**Sampling Theory:** Sampling, Sampling distributions, standard error, test of hypothesis for means and proportions, confidence limits for means, student’s t-distribution, Chi-square distribution as a test of goodness of fit. **Stochastic process:** Stochastic process, probability vector, stochastic matrices, fixed points, regular stochastic matrices, Markov chains, higher transition probability.

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

## Course Outcomes:

After studying this course, students will be able to:

- Solve first and second order ordinary differential equation arising in flow problems using single step and multistep numerical methods.
- Illustrate problems of potential theory, quantum mechanics and heat conduction by employing notions and properties of Bessel’s functions and Legendre’s polynomials.
- Explain the concepts of analytic functions, residues, poles of complex potentials and describe
conformal and Bilinear transformation arising in field theory and signal processing.

- Develop probability distribution of discrete, continuous random variables and joint probability distribution occurring in digital signal processing, information theory and design engineering.
- Demonstrate testing of hypothesis of sampling distributions and illustrate examples of Markov chains related to discrete parameter stochastic process.

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

### OBJECT ORIENTED CONCEPTS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

**SEMESTER – IV**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>IA Marks</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS42</td>
<td></td>
<td>40</td>
<td></td>
<td>60</td>
</tr>
</tbody>
</table>

**Total Number of Lecture Hours**: 40

**Exam Hours**: 03

**CREDITS – 03**

<table>
<thead>
<tr>
<th>Module 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction to Object Oriented Concepts:</strong></td>
<td>08 Hours</td>
</tr>
<tr>
<td>A Review of structures, Procedure–Oriented Programming system, Object Oriented Programming System, Comparison of Object Oriented Language with C, Console I/O, variables and reference variables, Function Prototyping, Function Overloading. <strong>Class and Objects:</strong> Introduction, member functions and data, objects and functions, objects and arrays, Namespaces, Nested classes, Constructors, Destructors.</td>
<td>Text book 1: Ch 1: 1.1 to 1.9 Ch 2: 2.1 to 2.6 Ch 4: 4.1 to 4.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction to Java:</strong> Java’s magic: the Byte code; Java Development Kit (JDK); the Java Buzzwords, Object-oriented programming; Simple Java programs. Data types, variables and arrays, Operators, Control Statements.</td>
<td>Text book 2: Ch:1 Ch: 2 Ch:3 Ch:4 Ch:5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classes, Inheritance, Exceptions, Packages and Interfaces:</strong> Classes: Classes fundamentals; Declaring objects; Constructors, this keyword, garbage collection. Inheritance: inheritance basics, using super, creating multi level hierarchy, method overriding. <strong>Exception handling:</strong> Exception handling in Java. Packages, Access Protection, Importing Packages, Interfaces.</td>
<td>Text book 2: Ch:6 Ch: 8 Ch:9 Ch:10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Multi Threaded Programming, Event Handling:</strong> 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, producer consumer problems. <strong>Event Handling:</strong> 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.</td>
<td>Text book 2: Ch 11: Ch: 22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Applet Class:</strong> Introduction, 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(); ApletContext and showDocument(); The AudioClip Interface; The AppletStub Interface;Output to the Console. <strong>Swings:</strong> 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.</td>
<td>Text book 2: Ch 21: Ch: 29 Ch: 30</td>
</tr>
</tbody>
</table>
### Course Outcomes:
After studying this course, students will be able to

- Explain the object-oriented concepts and JAVA.
- Develop computer programs to solve real world problems in Java.
- Develop simple GUI interfaces for a computer program to interact with users, and to comprehend the event-based GUI handling principles using Applets and swings.

### Question paper pattern:

The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

### Text Books:

   (Chapters 1, 2, 4)
   (Chapters 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 21, 22, 29, 30)

### Reference Book:


**Note:** Every institute shall organize a bridge organize on C++ either in the vacation or in the beginning of even semester.
<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Lecture Hours</th>
<th>Exam Hours</th>
<th>CREDITS – 04</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS43</td>
<td>40</td>
<td>60</td>
<td>50</td>
<td>03</td>
<td>04</td>
</tr>
</tbody>
</table>

**Module 1**

**Introduction:** What is an Algorithm? (T2:1.1), Algorithm Specification (T2:1.2), Analysis Framework (T1:2.1), **Performance Analysis:** Space complexity, Time complexity (T2:1.3). **Asymptotic Notations:** Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ), and Little-oh notation (o). Mathematical analysis of Non-Recursive and Recursive Algorithms with Examples (T1:2.2, 2.3, 2.4). **Important Problem Types:** Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. **Fundamental Data Structures:** Stacks, Queues, Graphs, Trees, Sets and Dictionaries. (T1:1.3,1.4)

**Module 2**

**Divide and Conquer:** General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum (T2:3.1, 3.3, 3.4), Merge sort, Quick sort (T1:4.1, 4.2), Strassen’s matrix multiplication (T2:3.8), Advantages and Disadvantages of divide and conquer. **Decrease and Conquer Approach:** Topological Sort. (T1:5.3)

**Module 3**

**Greedy Method:** General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines (T2:4.1, 4.3, 4.5). **Minimum cost spanning trees:** Prim’s Algorithm, Kruskal’s Algorithm (T1:9.1, 9.2). **Single source shortest paths:** Dijkstra's Algorithm (T1:9.3). **Optimal Tree problem:** Huffman Trees and Codes (T1:9.4). **Transform and Conquer Approach:** Heaps and Heap Sort (T1:6.4).

**Module 4**

**Dynamic Programming:** General method with Examples, Multistage Graphs (T2:5.1, 5.2). **Transitive Closure:** Warshall’s Algorithm, All Pairs Shortest Paths: Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem ((T1:8.2, 8.3, 8.4), Bellman-Ford Algorithm (T2:5.4), Travelling Sales Person problem (T2:5.9), Reliability design (T2:5.8).

**Module 5**

**Backtracking:** General method (T2:7.1), N-Queens problem (T1:12.1), Sum of subsets problem (T1:12.1), Graph coloring (T2:7.4), Hamiltonian cycles (T2:7.5). **Branch and Bound:** Assignment Problem, Travelling Sales Person problem (T1:12.2), 0/1 Knapsack problem (T2:8.2, T1:12.2): LC Branch and Bound solution (T2:8.2), FIFO Branch and Bound solution (T2:8.2). **NP-Complete and NP-Hard problems:** Basic concepts, non-deterministic algorithms, P, NP, NP-Complete, and NP-Hard classes (T2:11.1).

**Course Outcomes:** After studying this course, students will be able to

- Describe computational solution to well known problems like searching, sorting etc.
- Estimate the computational complexity of different algorithms.

10 Hours
- Develop an algorithm using appropriate design strategies for problem solving.

**Question paper pattern:**

The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

2. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education)
**Module 1**

**The x86 microprocessor:** Brief history of the x86 family, Inside the 8088/86, Introduction to assembly programming, Introduction to Program Segments, The Stack, Flag register, x86 Addressing Modes. **Assembly language programming:** Directives & a Sample Program, Assemble, Link & Run a program, More Sample programs, Control Transfer Instructions, Data Types and Data Definition, Full Segment Definition, Flowcharts and Pseudo code.

Text book 1: Ch 1: 1.1 to 1.7, Ch 2: 2.1 to 2.7

**Module 2**

**x86:** Instructions sets description, **Arithmetic and logic instructions and programs:** Unsigned Addition and Subtraction, Unsigned Multiplication and Division, Logic Instructions, BCD and ASCII conversion, Rotate Instructions. **INT 21H and INT 10H Programming:** Bios INT 10H Programming, DOS Interrupt 21H. 8088/86 Interrupts, x86 PC and Interrupt Assignment.

Text book 1: Ch 3: 3.1 to 3.5, Ch 4: 4.1, 4.2 Chapter 14: 14.1 and 14.2

**Module 3**

**Signed Numbers and Strings:** Signed number Arithmetic Operations, String operations. **Memory and Memory interfacing:** Memory address decoding, data integrity in RAM and ROM, 16-bit memory interfacing. **8255 I/O programming:** I/O addresses MAP of x86 PC’s, programming and interfacing the 8255.

Text book 1: Ch 6: 6.1, 6.2. Ch 10: 10.2, 10.4, 10.5. Ch 11: 11.1 to 11.4

**Module 4**

Microprocessors versus Microcontrollers, **ARM Embedded Systems:** The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software, **ARM Processor Fundamentals:** Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table, Core Extensions

Text book 2: Ch 1: 1.1 to 1.4, Ch 2: 2.1 to 2.5

**Module 5**

**Introduction to the ARM Instruction Set:** Data Processing Instructions, Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants, Simple programming exercises.

Text book 2: Ch 3: 3.1 to 3.6 (Excluding 3.5.2)

**Course Outcomes:** After studying this course, students will be able to

- Differentiate between microprocessors and microcontrollers
- Develop assembly language code to solve problems
- Explain interfacing of various devices to x86 family and ARM processor
- Demonstrate interrupt routines for interfacing devices

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

5. The Insider’s Guide to the ARM7 based microcontrollers, Hitex Ltd., 1st edition, 2005
**SOFTWARE ENGINEERING**
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)

**SEMESTER – IV**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS45</td>
<td>40</td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Processes: Models: Waterfall Model (Sec 2.1.1), Incremental Model (Sec 2.1.2) and Spiral Model (Sec 2.1.3). Process activities.</td>
<td></td>
</tr>
<tr>
<td>Requirements Engineering: Requirements Engineering Processes (Chap 4). Requirements Elicitation and Analysis (Sec 4.5). Functional and non-functional requirements (Sec 4.1). The software Requirements Document (Sec 4.2). Requirements Specification (Sec 4.3). Requirements validation (Sec 4.6). Requirements Management (Sec 4.7).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 2</th>
<th>11 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Models: Context models (Sec 5.1). Interaction models (Sec 5.2). Structural models (Sec 5.3). Behavioral models (Sec 5.4). Model-driven engineering (Sec 5.5).</td>
<td></td>
</tr>
<tr>
<td>Design and Implementation: Introduction to RUP (Sec 2.4). Design Principles (Chap 17). Object-oriented design using the UML (Sec 7.1). Design patterns (Sec 7.2). Implementation issues (Sec 7.3). Open source development (Sec 7.4).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 3</th>
<th>9 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Testing: Development testing (Sec 8.1). Test-driven development (Sec 8.2). Release testing (Sec 8.3). User testing (Sec 8.4). Test Automation (Page no 42, 70,212, 231,444,695).</td>
<td></td>
</tr>
<tr>
<td>Software Evolution: Evolution processes (Sec 9.1). Program evolution dynamics (Sec 9.2). Software maintenance (Sec 9.3). Legacy system management (Sec 9.4).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 4</th>
<th>10 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Planning: Software pricing (Sec 23.1). Plan-driven development (Sec 23.2). Project scheduling (Sec 23.3): Estimation techniques (Sec 23.5). Quality management: Software quality (Sec 24.1). Reviews and inspections (Sec 24.3). Software measurement and metrics (Sec 24.4). Software standards (Sec 24.2)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module 5</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Software Development: Coping with Change (Sec 2.3). The Agile Manifesto: Values and Principles. Agile methods: SCRUM (Ref “The SCRUM Primer, Ver 2.0”) and Extreme Programming (Sec 3.3). Plan-driven and agile development (Sec 3.2). Agile project management (Sec 3.4). Scaling agile methods (Sec 3.5):</td>
<td></td>
</tr>
</tbody>
</table>

**Course Outcomes:** After studying this course, students will be able to:
- Design a software system, component, or process to meet desired needs within realistic constraints.
- Assess professional and ethical responsibility
- Function on multi-disciplinary teams
- Make use of techniques, skills, and modern engineering tools necessary for engineering
practice
  • Comprehend software systems or parts of software systems.

**Question paper pattern:**

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

   (Listed topics only from Chapters 1,2,3,4, 5, 7, 8, 9, 23, and 24)

**Reference Books:**

2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India

**Web Reference for eBooks on Agile:**

# DATA COMMUNICATION

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)

## SEMESTER – IV

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS46</td>
<td>40</td>
<td>04</td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

## CREDITS – 04

<table>
<thead>
<tr>
<th>Contents Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module 1</strong></td>
</tr>
<tr>
<td><strong>Introduction:</strong> Data Communications, Networks, Network Types, Internet History, Standards and Administration, <strong>Networks Models:</strong> Protocol Layering, TCP/IP Protocol suite, The OSI model, <strong>Introduction to Physical Layer-1:</strong> Data and Signals, Digital Signals, Transmission Impairment, Data Rate limits, Performance, <strong>Digital Transmission:</strong> Digital to digital conversion (Only Line coding: Polar, Bipolar and Manchester coding).</td>
</tr>
<tr>
<td>10 Hours</td>
</tr>
<tr>
<td><strong>Module 2</strong></td>
</tr>
<tr>
<td><strong>Physical Layer-2:</strong> Analog to digital conversion (only PCM), Transmission Modes, <strong>Analog Transmission:</strong> Digital to analog conversion, <strong>Bandwidth Utilization:</strong> Multiplexing and Spread Spectrum, <strong>Switching:</strong> Introduction, Circuit Switched Networks and Packet switching.</td>
</tr>
<tr>
<td>10 Hours</td>
</tr>
<tr>
<td><strong>Module 3</strong></td>
</tr>
<tr>
<td><strong>Error Detection and Correction:</strong> Introduction, Block coding, Cyclic codes, Checksum, Forward error correction, <strong>Data link control:</strong> DLC services, Data link layer protocols, HDLC, and Point to Point protocol (Framing, Transition phases only).</td>
</tr>
<tr>
<td>10 Hours</td>
</tr>
<tr>
<td><strong>Module 4</strong></td>
</tr>
<tr>
<td><strong>Media Access control:</strong> Random Access, Controlled Access and Channelization, <strong>Wired LANs Ethernet:</strong> Ethernet Protocol, Standard Ethernet, Fast Ethernet, Gigabit Ethernet and 10 Gigabit Ethernet, <strong>Wireless LANs:</strong> Introduction, IEEE 802.11 Project and Bluetooth.</td>
</tr>
<tr>
<td>10 Hours</td>
</tr>
<tr>
<td><strong>Module 5</strong></td>
</tr>
<tr>
<td><strong>Other wireless Networks:</strong> WIMAX, Cellular Telephony, Satellite networks, <strong>Network layer Protocols :</strong> Internet Protocol, ICMPv4, Mobile IP, <strong>Next generation IP:</strong> IPv6 addressing, The IPv6 Protocol, The ICMPv6 Protocol and Transition from IPv4 to IPv6.</td>
</tr>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

## Course Outcomes:

After studying this course, students will be able to:

- Illustrate basic computer network technology.
- Identify the different types of network topologies and protocols.
- List and explain the layers of the OSI model and TCP/IP model.
- Comprehend the different types of network devices and their functions within a network
- Demonstrate subetting and routing mechanisms.

## Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.
<table>
<thead>
<tr>
<th><strong>Text Book:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Behrouz A. Forouzan, Data Communications and Networking 5E, 5th Edition, Tata McGraw-Hill, 2013. (Chapters 1.1 to 1.5, 2.1 to 2.3, 3.1, 3.3 to 3.6, 4.1 to 4.3, 5.1, 6.1, 6.2, 8.1 to 8.3, 10.1 to 10.5, 11.1 to 11.4, 12.1 to 12.3, 13.1 to 13.5, 15.1 to 15.3, 16.1 to 16.3, 19.1 to 19.3, 22.1 to 22.4)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Reference Books:</strong></th>
</tr>
</thead>
</table>
Design, develop, and implement the specified algorithms for the following problems using Java language under Linux/Windows environment. Netbeans/Eclipse IDE tool can be used for development and demonstration.

Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
</table>
| 1 | Create a Java class called `Student` with the following details as variables within it.  

(i) USN  
(ii) Name  
(iii) Branch  
(iv) Phone  

Write a Java program to create `nStudent` objects and print the USN, Name, Branch, and Phone of these objects with suitable headings. | Write a Java program to implement the Stack using arrays. Write Push(), Pop(), and Display() methods to demonstrate its working. |
| 2 | Design a superclass called `Staff` with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely `Teaching` (domain, publications), `Technical` (skills), and `Contract` (period). Write a Java program to read and display at least 3 `staff` objects of all three categories. | Write a Java class called `Customer` to store their name and date_of_birth. The date_of_birth format should be dd/mm/yyyy. Write methods to read customer data as `<name, dd/mm/yyyy>` and display as `<name, dd, mm, yyyy>` using StringTokenizer class considering the delimiter character as “/”. |
| 3 | Write a Java program to read two integers `a` and `b`. Compute `a/b` and print, when `b` is not zero. Raise an exception when `b` is equal to zero. | Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes the square of the number and prints; third thread will print the value of cube of the number. |
| 4 | Sort a given set of `n` integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of `n > 5000` and record the time taken to sort. Plot a graph of the time taken versus `n` on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case. | Sort a given set of `n` integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of `n > 5000`, and record the time taken to sort. Plot a graph of the time taken versus `n` on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case. |
and-conquer method works along with its time complexity analysis: worst case, average case and best case.

| 6  | Implement in Java, the **0/1 Knapsack** problem using (a) Dynamic Programming method (b) Greedy method. |
| 7  | From a given vertex in a weighted connected graph, find shortest paths to other vertices using **Dijkstra's algorithm**. Write the program in Java. |
| 8  | Find Minimum Cost Spanning Tree of a given connected undirected graph using **Kruskal's algorithm**. Use Union-Find algorithms in your program. |
| 9  | Find Minimum Cost Spanning Tree of a given connected undirected graph using **Prim's algorithm**. |
| 10 | Write Java programs to (a) Implement All-Pairs Shortest Paths problem using **Floyd's algorithm**. (b) Implement **Travelling Sales Person problem** using Dynamic programming. |
| 11 | Design and implement in Java to find a **subset** of a given set $S = \{S_1, S_2, \ldots, S_n\}$ of $n$ positive integers whose SUM is equal to a given positive integer $d$. For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$, there are two solutions $\{1,2,6\}$ and $\{1,8\}$. Display a suitable message, if the given problem instance doesn't have a solution. |
| 12 | Design and implement in Java to find all **Hamiltonian Cycles** in a connected undirected Graph $G$ of $n$ vertices using backtracking principle. |

**Course Outcomes:** The students should be able to:

- Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)
- Develop variety of algorithms such as sorting, graph related, combinatorial, etc., in a high level language.
- Analyze and compare the performance of algorithms using language features.
- Apply and implement learned algorithm design techniques and data structures to solve real-world problems.

**Conduction of Practical Examination:**
All laboratory experiments (Twelve problems) are to be included for practical examination. Students are allowed to pick one experiment from the lot.
To generate the data set use random number generator function.
Strictly follow the instructions as printed on the cover page of answer script for breakup of marks
**Marks distribution:** Procedure + Conduction + Viva: 15 + 70 + 15 (100). Change of experiment is allowed only once and marks allotted to the procedure
# MICROPROCESSOR AND MICROCONTROLLER LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

**SEMESTER – IV**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CSL48</td>
<td></td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 02**

**Description**

Demonstration and Explanation hardware components and Faculty in-charge should explain 8086 architecture, pin diagram in one slot. The second slot, the Faculty in-charge should explain instruction set types/category etc. Students have to prepare a write-up on the same and include it in the Lab record and to be evaluated.

Laboratory Session-1: Write-up on Microprocessors, 8086 Functional block diagram, Pin diagram and description. The same information is also taught in theory class; this helps the students to understand better.

Laboratory Session-2: Write-up on Instruction group, Timing diagrams, etc. The same information is also taught in theory class; this helps the students to understand better.

**Note:** These TWO Laboratory sessions are used to fill the gap between theory classes and practical sessions. Both sessions are evaluated as lab experiments for 20 marks.

**Experiments**

- Develop and execute the following programs using 8086 Assembly Language. Any suitable assembler like MASM/TASM/8086 kit or any equivalent software may be used.
- Program should have suitable comments.
- The board layout and the circuit diagram of the interface are to be provided to the student during the examination.
- Software Required: Open source ARM Development platform, KEIL IDE and Proteus for simulation

**SOFTWARE PROGRAMS: PART A**

1. Design and develop an assembly language program to search a key element “X” in a list of ‘n’ 16-bit numbers. Adopt Binary search algorithm in your program for searching.
2. Design and develop an assembly program to sort a given set of ‘n’ 16-bit numbers in ascending order. Adopt Bubble Sort algorithm to sort given elements.
3. Develop an assembly language program to reverse a given string and verify whether it is a palindrome or not. Display the appropriate message.
4. Develop an assembly language program to compute nCr using recursive procedure. Assume that ‘n’ and ‘r’ are non-negative integers.
5. Design and develop an assembly language program to read the current time and Date from the system and display it in the standard format on the screen.
6. To write and simulate ARM assembly language programs for data transfer, arithmetic and logical operations (Demonstrate with the help of a suitable program).
7. To write and simulate C Programs for ARM microprocessor using KEIL (Demonstrate with the help of a suitable program)

**Note:** To use KEIL one may refer the book:  *Insider’s Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005*
### HARDWARE PROGRAMS: PART B

8. **a.** Design and develop an assembly program to demonstrate BCD Up-Down Counter (00-99) on the Logic Controller Interface.  
   **b.** Design and develop an assembly program to read the status of two 8-bit inputs (X & Y) from the Logic Controller Interface and display X*Y.  

9. Design and develop an assembly program to display messages “FIRE” and “HELP” alternately with flickering effects on a 7-segment display interface for a suitable period of time. Ensure a flashing rate that makes it easy to read both the messages (Examiner does not specify these delay values nor is it necessary for the student to compute these values).  

10. Design and develop an assembly program to drive a Stepper Motor interface and rotate the motor in specified direction (clockwise or counter-clockwise) by N steps (Direction and N are specified by the examiner). Introduce suitable delay between successive steps. (Any arbitrary value for the delay may be assumed by the student).  

11. Design and develop an assembly language program to  
   **a.** Generate the Sine Wave using DAC interface (The output of the DAC is to be displayed on the CRO).  
   **b.** Generate a Half Rectified Sine waveform using the DAC interface. (The output of the DAC is to be displayed on the CRO).  

12. To interface LCD with ARM processor-- ARM7TDMI/LPC2148. Write and execute programs in C language for displaying text messages and numbers on LCD  

13. To interface Stepper motor with ARM processor-- ARM7TDMI/LPC2148. Write a program to rotate stepper motor  

### Study Experiments:  
1. Interfacing of temperature sensor with ARM freedom board (or any other ARM microprocessor board) and display temperature on LCD  
2. To design ARM cortex based automatic number plate recognition system  
3. To design ARM based power saving system  

### Course Outcomes: After studying this course, students will be able to  
- Summarize 80x86 instruction sets and comprehend the knowledge of how assembly language works.  
- Design and develop assembly programs using 80x86 assembly language instructions  
- Infer functioning of hardware devices and interfacing them to x86 family  
- Choose processors for various kinds of applications.  

### Conduction of Practical Examination:  
- All laboratory experiments (all 7 + 6 nos) are to be included for practical examination.  
- Students are allowed to pick one experiment from each of the lot.  
- Strictly follow the instructions as printed on the cover page of answer script for breakup of marks  
- **PART –A: Procedure + Conduction + Viva:** 08 + 35 + 07 (50)  
- **PART –B: Procedure + Conduction + Viva:** 08 + 35 + 07 (50)  
- Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.
## MANAGEMENT AND ENTREPRENEURSHIP FOR IT INDUSTRY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

### SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS51</td>
<td>40</td>
<td>4</td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

### Module – 1

**Introduction** - Meaning, nature and characteristics of management, scope and Functional areas of management, goals of management, levels of management, brief overview of evolution of management theories. Planning- Nature, importance, types of plans, steps in planning, Organizing- nature and purpose, types of Organization, Staffing- meaning, process of recruitment and selection

- **Teaching Hours**: 10 Hours

### Module – 2

**Directing and controlling** - meaning and nature of directing, leadership styles, motivation Theories, Communication- Meaning and importance, Coordination- meaning and importance, Controlling- meaning, steps in controlling, methods of establishing control.

- **Teaching Hours**: 10 Hours

### Module – 3

**Entrepreneur** – meaning of entrepreneur, characteristics of entrepreneurs, classification and types of entrepreneurs, various stages in entrepreneurial process, role of entrepreneurs in economic development, entrepreneurship in India and barriers to entrepreneurship. Identification of business opportunities, market feasibility study, technical feasibility study, financial feasibility study and social feasibility study.

- **Teaching Hours**: 10 Hours

### Module – 4

**Preparation of project and ERP** - meaning of project, project identification, project selection, project report, need and significance of project report, contents, formulation, guidelines by planning commission for project report, Enterprise Resource Planning: Meaning and Importance- ERP and Functional areas of Management – Marketing / Sales- Supply Chain Management – Finance and Accounting – Human Resources – Types of reports and methods of report generation

- **Teaching Hours**: 10 Hours

### Module – 5

**Micro and Small Enterprises**: Definition of micro and small enterprises, characteristics and advantages of micro and small enterprises, steps in establishing micro and small enterprises, Government of India industrial policy 2007 on micro and small enterprises, case study (Microsoft), Case study(Captain G R Gopinath), case study (N R Narayana Murthy & Infosys), **Institutional support**: MSME-DI, NSIC, SIDBI, KIADB, KSSIDC, TECSOK, KSFC, DIC and District level single window agency, **Introduction to IPR**.

- **Teaching Hours**: 10 Hours

### Course outcomes: The students should be able to:

- Define management, organization, entrepreneur, planning, staffing, ERP and outline their importance in entrepreneurship
- Utilize the resources available effectively through ERP
- Make use of IPRs and institutional support in entrepreneurship

### Question paper pattern:
The question paper will have TEN questions. 
There will be TWO questions from each module.  
Each question will have questions covering all the topics under a module.  
The students will have to answer FIVE full questions, selecting ONE full question from each module.

<table>
<thead>
<tr>
<th>Text Books:</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Reference Books:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Entrepreneurship Development -S S Khanka -S Chand &amp; Co.</td>
<td></td>
</tr>
</tbody>
</table>
# COMPUTER NETWORKS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

## SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>CREDITS – 04</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS52</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

### Module – 1


**T1: Chap 2**

### Module – 2


**T1: Chap 3**

### Module – 3


**T1: Chap 4: 4.3-4.7**

### Module – 4


**T1: Chap 4**
Addressing, Routing to a mobile node, Mobile IP, Managing mobility in cellular Networks, Routing calls to a Mobile user, Handoffs in GSM, Wireless and Mobility: Impact on Higher-layer protocols.

**T1: Chap: 6 : 6.4-6.8**

**Module – 5**


**Network Support for Multimedia:** Quality-of-Service (QoS) Guarantees: Resource Reservation and Call Admission

**T1: Chap: 7**

**Course outcomes:** The students should be able to:

- Explain principles of application layer protocols
- Outline transport layer services and infer UDP and TCP protocols
- Classify routers, IP and Routing Algorithms in network layer
- Explain the Wireless and Mobile Networks covering IEEE 802.11 Standard
- Define Multimedia Networking and Network Management

**Question paper pattern:**

The question paper will have TEN questions.

There will be TWO questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**


**Reference Books:**

2. Larry L Peterson and Brusce S Davie, Computer Networks, fifth edition, ELSEVIER
3. Andrew S Tanenbaum, Computer Networks, fifth edition, Pearson
DATABASE MANAGEMENT SYSTEM
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS53</td>
<td>40</td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

CREDITS – 04

Module – 1

Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications.
Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment. Conceptual Data Modelling using Entities and Relationships: Entity types, Entity sets, attributes, roles, and structural constraints, Weak entity types, ER diagrams, examples, Specialization and Generalization.

Textbook 1: Ch 1.1 to 1.8, 2.1 to 2.6, 3.1 to 3.10

Module – 2

Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations. Relational Algebra: Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra. Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relational mapping. SQL: SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL.

Textbook 1: Ch4.1 to 4.5, 5.1 to 5.3, 6.1 to 6.5, 8.1; Textbook 2: 3.5

Module – 3

SQL : Advances Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL, Schema change statements in SQL. Database Application Development: Accessing databases from applications, An introduction to JDBC, JDBC classes and interfaces, SQLJ, Stored procedures, Case study: The internet Bookshop. Internet Applications: The three-Tier application architecture, The presentation layer, The Middle Tier

Textbook 1: Ch7.1 to 7.4; Textbook 2: 6.1 to 6.6, 7.5 to 7.7.

Module – 4


Teaching Hours

10 Hours

10 Hours

10 Hours

10 Hours
<table>
<thead>
<tr>
<th>Textbook 1: Ch14.1 to 14.7, 15.1 to 15.6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module – 5</strong></td>
</tr>
<tr>
<td><strong>Transaction Processing:</strong> Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL. <strong>Concurrency Control in Databases:</strong> Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking. <strong>Introduction to Database Recovery Protocols:</strong> Recovery Concepts, NO-UNDO/REDO recovery based on Deferred update, Recovery techniques based on immediate update, Shadow paging, Database backup and recovery from catastrophic failures</td>
</tr>
</tbody>
</table>

**Textbook 1: 20.1 to 20.6, 21.1 to 21.7, 22.1 to 22.4, 22.7.**

**Course outcomes:** The students should be able to:

- Summarize the concepts of database objects; enforce integrity constraints on a database using RDBMS.
- Use Structured Query Language (SQL) for database manipulation.
- Design simple database systems
- Design code for some application to interact with databases.

**Question paper pattern:**

The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**


**Reference Books:**

### AUTOMATA THEORY AND COMPUTABILITY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

#### SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS54</td>
<td></td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why study the Theory of Computation, Languages and Strings: Strings, Languages. A Language Hierarchy, Computation, <strong>Finite State Machines (FSM):</strong> Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, From FSMs to Operational Systems, Simulators for FSMs, Minimizing FSMs, Canonical form of Regular languages, Finite State Transducers, Bidirectional Transducers. <strong>Textbook 1:</strong> Ch 1, 2, 3, 4, 5.1 to 5.10</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular Expressions (RE): what is a RE?, Kleene’s theorem, Applications of REs, Manipulating and Simplifying REs. <strong>Regular Grammars:</strong> Definition, Regular Grammars and Regular languages. <strong>Regular Languages (RL) and Non-regular Languages:</strong> How many RLs, To show that a language is regular, Closure properties of RLs, to show some languages are not RLs. <strong>Textbook 1:</strong> Ch 6, 7, 8: 6.1 to 6.4, 7.1, 7.2, 8.1 to 8.4</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Free Grammars(CFG): Introduction to Rewrite Systems and Grammars, CFGs and languages, designing CFGs, simplifying CFGs, proving that a Grammar is correct, Derivation and Parse trees, Ambiguity, Normal Forms. <strong>Pushdown Automata (PDA):</strong> Definition of non-deterministic PDA, Deterministic and Non-deterministic PDAs, Non-determinism and Halting, alternative equivalent definitions of a PDA, alternatives that are not equivalent to PDA. <strong>Textbook 1:</strong> Ch 11, 12: 11.1 to 11.8, 12.1, 12.2, 12.4, 12.5, 12.6</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Free and Non-Context-Free Languages: Where do the Context-Free Languages(CFL) fit, Showing a language is context-free, Pumping theorem for CFL, Important closure properties of CFLs, Deterministic CFLs. Algorithms and Decision Procedures for CFLs: Decidable questions, Un-decidable questions. <strong>Turing Machine:</strong> Turing machine model, Representation, Language acceptability by TM, design of TM, Techniques for TM construction. <strong>Textbook 1:</strong> Ch 13: 13.1 to 13.5, Ch 14: 14.1, 14.2, <strong>Textbook 2:</strong> Ch 9.1 to 9.6</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variants of Turing Machines (TM), The model of Linear Bounded automata: Decidability: Definition of an algorithm, decidability, decidable languages, Undecidable languages, halting problem of TM, Post correspondence problem. <strong>Complexity:</strong> Growth rate of functions, the classes of P and NP, Quantum Computation: quantum computers, Church-Turing thesis. <strong>Textbook 2:</strong> Ch 9.7 to 9.8, 10.1 to 10.7, 12.1, 12.2, 12.8, 12.8.1, 12.8.2</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

**Course outcomes:** The students should be able to:
- Tell the core concepts in automata theory and Theory of Computation
- Explain how to translate between different models of Computation (e.g., Deterministic and Non-deterministic and Software models).
- Interpret Grammars and Automata (recognizers) for different language classes and become knowledgeable about restricted models of Computation (Regular, Context Free) and their relative powers.
- Develop skills in formal reasoning and reduction of a problem to a formal model, with an emphasis on semantic precision and conciseness.
- Classify a problem with respect to different models of Computation.

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
5. Basavaraj S. Anami, Karibasappa K G, Formal Languages and Automata theory, Wiley India, 2012
# OBJECT ORIENTED MODELING AND DESIGN

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Total Number of Lecture Hours</th>
<th>CREDITS – 03</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS551</td>
<td>40</td>
<td>60</td>
<td>3</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

## Module – 1

**Introduction, Modelling Concepts and Class Modelling:**
What is Object orientation? What is OO development? OO Themes; Evidence for usefulness of OO development; OO modelling history. Modelling as Design technique: Modelling; abstraction; The Three models. Class Modelling: Object and Class Concept; Link and associations concepts; Generalization and Inheritance; A sample class model; Navigation of class models; Advanced Class Modelling, Advanced object and class concepts; Association ends; N-ary associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived Data; Packages.

**Text Book-1:** Ch 1, 2, 3 and 4

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

## Module – 2

**UseCase Modelling and Detailed Requirements:**
Overview; Detailed object-oriented Requirements definitions; System Processes-A use case/Scenario view; Identifying Input and outputs-The System sequence diagram; Identifying Object Behaviour-The state chart Diagram; Integrated Object-oriented Models.

**Text Book-2:** Chapter- 6:Page 210 to 250

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

## Module – 3

**Process Overview, System Conception and Domain Analysis:**
Process Overview: Development stages; Development life Cycle; 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.

**Text Book-1:** Chapter- 10,11, and 12

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

## Module – 4

**Use case Realization :The Design Discipline within up iterations:**
Object Oriented Design-The Bridge between Requirements and Implementation; Design Classes and Design within Class Diagrams; Interaction Diagrams-Realizing Use Case and defining methods; Designing with Communication Diagrams; Updating the Design Class Diagram; Package Diagrams-Structuring the Major Components; Implementation Issues for Three-Layer Design.

**Text Book-2:** Chapter 8: page 292 to 346

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

## Module – 5

**Design Patterns:**
Introduction; what is a design pattern?, Describing design patterns, the catalogue of design patterns, Organizing the catalogue, How design patterns solve design problems, how to select a design patterns, how to use a design pattern; Creational patterns: prototype and singleton (only); structural patterns adaptor and proxy (only).

**Text Book-3:** Ch-1: 1.1, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, Ch-3, Ch-4.

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

## Course outcomes:
The students should be able to:
- Describe the concepts of object-oriented and basic class modelling.
- Draw class diagrams, sequence diagrams and interaction diagrams to solve problems.
- Choose and apply a befitting design pattern for the given problem.

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
## INTRODUCTION TO SOFTWARE TESTING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

### SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Test Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS552</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

### Number of Lecture Hours/Week

<table>
<thead>
<tr>
<th>Total Number of Lecture Hours</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>60</td>
<td>03</td>
</tr>
</tbody>
</table>

### CREDITS – 03

### Module – 1

**Basics of Software Testing:** Basic definitions, Software Quality, Requirements, Behaviour and Correctness, Correctness versus Reliability, Testing and Debugging, Test cases, Insights from a Venn diagram, Identifying test cases, Test-generation Strategies, Test Metrics, Error and fault taxonomies, Levels of testing, Testing and Verification, Static Testing.

**Textbook 3:** Ch 1:1.2 - 1.5, 3; **Textbook 1:** Ch 1

### Module – 2

**Problem Statements:** Generalized pseudo code, the triangle problem, the NextDate function, the commission problem, the SATM (Simple Automatic Teller Machine) problem, the currency converter, Saturn windshield wiper

**Functional Testing:** Boundary value analysis, Robustness testing, Worst-case testing, Robust Worst testing for triangle problem, NextDate problem and commission problem, Equivalence classes, Equivalence test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations, Decision tables, Test cases for the triangle problem, NextDate function, and the commission problem, Guidelines and observations.

**Textbook 1:** Ch 2, 5, 6 & 7, **Textbook 2:** Ch 3

### Module – 3

**Fault Based Testing:** Overview, Assumptions in fault based testing, Mutation analysis, Fault-based adequacy criteria, Variations on mutation analysis.

**Structural Testing:** Overview, Statement testing, Branch testing, Condition testing, Path testing: DD paths, Test coverage metrics, Basis path testing, guidelines and observations, Data –Flow testing: Definition–Use testing, Slice-based testing, Guidelines and observations.

**T2:** Chapter 16, 12  **T1:** Chapter 9 & 10

### Module – 4

**Test Execution:** Overview of test execution, from test case specification to test cases, Scaffolding, Generic versus specific scaffolding, Test oracles, Self-checks as oracles, Capture and replay **Process Framework:** Basic principles: Sensitivity, redundancy, restriction, partition, visibility, Feedback, the quality process, Planning and monitoring, Quality goals, Dependability properties, Analysis Testing, Improving the process, Organizational factors.

**Planning and Monitoring the Process:** Quality and process, Test and analysis strategies and plans, Risk planning, monitoring the process, Improving the process, the quality team.

**T2:** Chapter 17, 20.

### Module – 5

**Integration and Component-Based Software Testing:** Overview, Integration testing strategies, Testing components and assemblies. System, Acceptance and

---

**Teaching Hours**

<table>
<thead>
<tr>
<th>Basics of Software Testing</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Statements</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Functional Testing</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Fault Based Testing</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Structural Testing</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Test Execution</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Integration and Component-Based Software Testing</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>
Regression Testing: Overview, System testing, Acceptance testing, Usability, Regression testing, Regression test selection techniques, Test case prioritization and selective execution. **Levels of Testing, Integration Testing:** Traditional view of testing levels, Alternative life-cycle models, The SATM system, Separating integration and system testing. A closer look at the SATM system, Decomposition-based, call graph-based, Path-based integrations.

**T2: Chapter 21 & 22, T1 : Chapter 12 & 13**

**Course outcomes:** The students should be able to:

- Identify test cases for any given problem.
- Compare the different testing techniques.
- Classify the problems according to a suitable testing model.
- Apply the appropriate technique for the design of flow graph.
- Create appropriate document for the software artefact.

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**


**Reference Books:**

# ADVANCED JAVA AND J2EE

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

## SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS553</td>
<td>40</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

### Module – 1

Enumerations, Autoboxing and Annotations(metadata):
Enumerations, Enumeration fundamentals, the values() and valueOf() Methods, java enumerations are class types, enumerations Inherit Enum, example, type wrappers, Autoboxing, Autoboxing and Methods, Autoboxing/Unboxing occurs in Expressions, Autoboxing/Unboxing, Boolean and character values, Autoboxing/Unboxing helps prevent errors, A word of Warning. Annotations, Annotation basics, specifying retention policy, Obtaining Annotations at run time by use of reflection, Annotated element Interface, Using Default values, Marker Annotations, Single Member annotations, Built-In annotations.

**Teaching Hours**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module – 2

The collections and Framework:

**Teaching Hours**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>8 Hours</th>
</tr>
</thead>
</table>

### Module – 3

String Handling:
The String Constructors, String Length, Special String Operations, String Literals, String Concatenation, String Conversion and toString( ) Character Extraction, charAt( ), getChars( ), getBytes( ) toCharArray(), String Comparison, equals( ) and equalsIgnoreCase( ), regionMatches( ) startsWith( ) and endsWith( ), equals( ) Versus == , compareTo( ) Searching Strings, Modifying a String, substring( ), concat( ), replace( ), trim( ), Data Conversion Using valueOf( ), Changing the Case of Characters Within a String, Additional String Methods, StringBuffer , StringBuffer Constructors, length( ) and capacity( ), ensureCapacity( ), setLength( ), charAt( ) and setCharAt( ), getChars( ),append( ), insert( ), reverse( ), delete( ) and deleteCharAt( ), replace( ), substring( ), Additional StringBuffer Methods, StringBuilder

**Text Book 1: Ch 15**

### Module – 4

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

**Teaching Hours**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Book 1: Ch 31</td>
<td>Text Book 2: Ch 11</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td><strong>Module – 5</strong></td>
<td><strong>8 Hours</strong></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

**Text Book 2: Ch 06**

**Course outcomes:** The students should be able to:

- Interpret the need for advanced Java concepts like enumerations and collections in developing modular and efficient programs
- Build client-server applications and TCP/IP socket programs
- Illustrate database access and details for managing information using the JDBC API
- Describe how servlets fit into Java-based web application architecture
- Develop reusable software components using Java Beans

**Question paper pattern:**
The question paper will have TEN questions. There will be TWO questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
# ADVANCED ALGORITHMS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS554</td>
<td>40</td>
<td>3</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis Techniques: Growth functions, Recurrences and solution of recurrence equations; Amortized analysis: Aggregate, Accounting, and Potential methods, String Matching Algorithms: Naive Algorithm; Robin-Karp Algorithm, String matching with Finite Automata, Knuth-Morris-Pratt and Boyer-Moore Algorithms</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Theoretic Algorithms: Elementary notions, GCD, Modular arithmetic, Solving modular linear equations, The Chinese remainder theorem, Powers of an element RSA Cryptosystem, Primality testing, Integer factorization, - Huffman Codes, Polynomials. FFT-Huffman codes: Concepts, construction, Proof correctness of Huffman's algorithm; Representation of polynomials</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFT and FFT efficient implementation of FFT, Graph Algorithms, Bellman-Ford Algorithm Shortest paths in a DAG, Johnson's Algorithm for sparse graphs, Flow networks and the Ford-Fulkerson Algorithm, Maximum bipartite matching.</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational Geometry-I: Geometric data structures using, C, Vectors, Points, Polygons, Edges Geometric objects in space; Finding the intersection of a line and a triangle, Finding star-shaped polygons using incremental insertion.</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational Geometry-II: Clipping: Cyrus-Beck and Sutherland-Hodman Algorithms; Triangulating, monotonic polygons; Convex hulls, Gift wrapping and Graham Scan; Removing hidden surfaces</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

**Course outcomes:** The students should be able to:

- Explain the principles of algorithms analysis approaches
- Apply different theoretic based strategies to solve problems
- Illustrate the complex signals and data flow in networks with usage of tools
- Describe the computational geometry criteria.

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

<table>
<thead>
<tr>
<th>Reference Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject Code</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>17CS561</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java’s Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions. <strong>Text book 1: Ch 9, Ch 10</strong></td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enumerations, Type Wrappers, I/O, Applets, and Other Topics: I/O Basics, Reading Console Input, Writing Console Output, The PrintWriter Class, Reading and Writing Files, Applet Fundamentals, The transient and volatile Modifiers, Using instanceof, strictfp, Native Methods, Using assert, Static Import, Invoking Overloaded Constructors Through this(), String Handling: The String</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>
Constructors, String Length, Special String Operations, Character Extraction, String Comparison, Searching Strings, Modifying a String, Data Conversion Using `valueOf()`, Changing the Case of Characters Within a String, Additional String Methods, `StringBuilder`, `StringBuffer`.

**Text book 1: Ch 12.1, 12.2, Ch 13, Ch 15**

**Course outcomes:** The students should be able to:
- Explain the object-oriented concepts and JAVA.
- Develop computer programs to solve real world problems in Java.
- Develop simple GUI interfaces for a computer program to interact with users

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
# ARTIFICIAL INTELLIGENCE

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

## SEMESTER – V

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS562</td>
<td></td>
<td>60</td>
<td>40</td>
<td>3</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

## Module – 1

What is artificial intelligence?, Problems, Problem Spaces and search, Heuristic search technique

**TextBook1**: Ch 1, 2 and 3

### Teaching Hours

| 8 Hours |

## Module – 2

Knowledge Representation Issues, Using Predicate Logic, Representing knowledge using Rules,

**TextBook1**: Ch 4, 5 and 6.

### Teaching Hours

| 8 Hours |

## Module – 3

Symbolic Reasoning under Uncertainty, Statistical reasoning, Weak Slot and Filter Structures.

**TextBook1**: Ch 7, 8 and 9.

### Teaching Hours

| 8 Hours |

## Module – 4

Strong slot-and-filler structures, Game Playing.

**TextBook1**: Ch 10 and 12

### Teaching Hours

| 8 Hours |

## Module – 5

Natural Language Processing, Learning, Expert Systems.

**TextBook1**: Ch 15, 17 and 20

### Teaching Hours

| 8 Hours |

## Course outcomes:
The students should be able to:

- Identify the AI based problems
- Apply techniques to solve the AI problems
- Define learning and explain various learning techniques
- Discuss expert systems

## Question paper pattern:
The question paper will have TEN questions.
- There will be TWO questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer FIVE full questions, selecting ONE full question from each module.

## Text Books:


## Reference Books:

2. Dan W. Patterson, Introduction to Artificial Intelligence and Expert Systems – Prentice Hal of India.
### EMBEDDED SYSTEMS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS563</td>
<td>40</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module – 1</td>
<td></td>
</tr>
<tr>
<td><strong>Introduction to embedded systems:</strong> Embedded systems, Processor embedded into a system, Embedded hardware units and device in a system, Embedded software in a system, Examples of embedded systems, Design process in embedded system, Formalization of system design, Design process and design examples, Classification of embedded systems, skills required for an embedded system designer.</td>
<td></td>
</tr>
<tr>
<td><strong>Module – 2</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Devices and communication buses for devices network:</strong> IO types and example, Serial communication devices, Parallel device ports, Sophisticated interfacing features in device ports, Wireless devices, Timer and counting devices, Watchdog timer, Real time clock, Networked embedded systems, Serial bus communication protocols, Parallel bus device protocols-parallel communication internet using ISA, PCI, PCI-X and advanced buses, Internet enabled systems-network protocols, Wireless and mobile system protocols.</td>
<td></td>
</tr>
<tr>
<td><strong>Module – 3</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Device drivers and interrupts and service mechanism:</strong> Programming-I/O busy-wait approach without interrupt service mechanism, ISR concept, Interrupt sources, Interrupt servicing (Handling) Mechanism, Multiple interrupts, Context and the periods for context switching, interrupt latency and deadline, Classification of processors interrupt service mechanism from Context-saving angle, Direct memory access, Device driver programming.</td>
<td></td>
</tr>
<tr>
<td><strong>Module – 4</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Inter process communication and synchronization of processes, Threads and tasks:</strong> Multiple process in an application, Multiple threads in an application, Tasks, Task states, Task and Data, Clear-cut distinction between functions. ISRS and tasks by their characteristics, concept and semaphores, Shared data, Inter-process communication, Signal function, Semaphore functions, Message Queue functions, Mailbox functions, Pipe functions, Socket functions, RPC functions.</td>
<td></td>
</tr>
<tr>
<td><strong>Module – 5</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Real-time operating systems:</strong> OS Services, Process management, Timer functions, Event functions, Memory management, Device, file and IO subsystems management, Interrupt routines in RTOS environment and handling of interrupt source calls, Real-time operating systems, Basic design using an RTOS, RTOS task scheduling models, interrupt latency and response of the tasks as performance metrics, OS security issues. Introduction to embedded software development process and tools, Host and target machines, Linking and location software.</td>
<td></td>
</tr>
</tbody>
</table>

**Course outcomes:** The students should be able to:
- Distinguish the characteristics of embedded computer systems.
- Identify the various vulnerabilities of embedded computer systems.
- Design and develop modules using RTOS.
- Explain RPC, threads and tasks

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
DOT NET FRAMEWORK FOR APPLICATION DEVELOPMENT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

**SEMESTER – V**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS564</td>
<td>40</td>
<td>3</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing Microsoft Visual C# and Microsoft Visual Studio 2015: Welcome to C#, Working with variables, operators and expressions, Writing methods and applying scope, Using decision statements, Using compound assignment and iteration statements, Managing errors and exceptions T1: Chapter 1 – Chapter 6</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding the C# object model: Creating and Managing classes and objects, Understanding values and references, Creating value types with enumerations and structures, Using arrays Textbook 1: Ch 7 to 10</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding parameter arrays, Working with inheritance, Creating interfaces and defining abstract classes, Using garbage collection and resource management Textbook 1: Ch 11 to 14</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Extensible Types with C#: Implementing properties to access fields, Using indexers, Introducing generics, Using collections Textbook 1: Ch 15 to 18</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enumerating Collections, Decoupling application logic and handling events, Querying in-memory data by using query expressions, Operator overloading Textbook 1: Ch 19 to 22</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

**Course outcomes:** The students should be able to:
- Build applications on Visual Studio .NET platform by understanding the syntax and semantics of C#
- Demonstrate Object Oriented Programming concepts in C# programming language
- Design custom interfaces for applications and leverage the available built-in interfaces in building complex applications.
- Illustrate the use of generics and collections in C#
- Compose queries to query in-memory data and define own operator behaviour

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**
Reference Books:

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>17CS565</th>
<th>IA Marks</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>3</td>
<td>Exam Marks</td>
<td>60</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction, Cloud Computing at a Glance, The Vision of Cloud Computing,</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Defining a Cloud, A Closer Look, Cloud Computing Reference Model,</td>
<td></td>
</tr>
<tr>
<td>Characteristics and Benefits, Challenges Ahead, Historical Developments,</td>
<td></td>
</tr>
<tr>
<td>Distributed Systems, Virtualization, Web 2.0, Service-Oriented Computing,</td>
<td></td>
</tr>
<tr>
<td>Utility-Oriented Computing, Building Cloud Computing Environments,</td>
<td></td>
</tr>
<tr>
<td>Application Development, Infrastructure and System Development, Computing</td>
<td></td>
</tr>
<tr>
<td>Platforms and Technologies, Amazon Web Services (AWS), Google AppEngine,</td>
<td></td>
</tr>
<tr>
<td>Microsoft Azure, Hadoop, Force.com and Salesforce.com, Manjrasoft Aneka</td>
<td></td>
</tr>
<tr>
<td>Virtualization, Introduction, Characteristics of Virtualized, Environments</td>
<td></td>
</tr>
<tr>
<td>Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types</td>
<td></td>
</tr>
<tr>
<td>of Virtualization, Virtualization and Cloud Computing, Pros and Cons of</td>
<td></td>
</tr>
<tr>
<td>Virtualization, Technology</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Computing Architecture, Introduction, Cloud Reference Model,</td>
<td></td>
</tr>
<tr>
<td>Architecture, Infrastructure / Hardware as a Service, Platform as a Service,</td>
<td></td>
</tr>
<tr>
<td>Software as a Service, Types of Clouds, Public Clouds, Private Clouds,</td>
<td></td>
</tr>
<tr>
<td>Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges,</td>
<td></td>
</tr>
<tr>
<td>Cloud Definition, Cloud Interoperability and Standards Scalability and</td>
<td></td>
</tr>
<tr>
<td>Fault Tolerance Security, Trust, and Privacy Organizational Aspects</td>
<td></td>
</tr>
<tr>
<td>Aneka: Cloud Application Platform, Framework Overview, Anatomy of the</td>
<td></td>
</tr>
<tr>
<td>Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric</td>
<td></td>
</tr>
<tr>
<td>Services, foundation Services, Application Services, Building Aneka Clouds,</td>
<td></td>
</tr>
<tr>
<td>Infrastructure Organization, Logical Organization, Private Cloud Deployment</td>
<td></td>
</tr>
<tr>
<td>Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud</td>
<td></td>
</tr>
<tr>
<td>Programming and Management, Aneka SDK, Management Tools</td>
<td></td>
</tr>
</tbody>
</table>

<p>| Module – 3                                                                 | 8 Hours        |
| Concurrent Computing: Thread Programming, Introducing Parallelism for Single|                |
| Machine Computation, Programming Applications with Threads, What is a      |                |
| Thread?, Thread APIs, Techniques for Parallel Computation with Threads,    |                |
| Multithreading with Aneka, Introducing the Thread Programming Model, Aneka  |                |
| Thread vs. Common Threads, Programming Applications with Aneka Threads,    |                |
| Aneka Threads Application Model, Domain Decomposition: Matrix Multiplication|                |
| Functional Decomposition: Sine, Cosine, and Tangent. High-Throughput        |                |
| Computing: Task Programming, Task Computing, Characterizing a Task,        |                |
| Computing Categories, Frameworks for Task Computing, Task-based Application|                |
| Models, Embarrassingly Parallel Applications, Parameter Sweep Applications,|                |
| MPI Applications, Workflow Applications with Task Dependencies, Aneka      |                |
| Task-Based Programming, Task Programming                                  |                |</p>
<table>
<thead>
<tr>
<th>Model, Developing Applications with the Task Model, Developing Parameter Sweep Application, Managing Workflows.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module – 4</strong></td>
</tr>
<tr>
<td><strong>Module – 5</strong></td>
</tr>
<tr>
<td><strong>Course outcomes:</strong> The students should be able to:</td>
</tr>
<tr>
<td>• Explain the concepts and terminologies of cloud computing</td>
</tr>
<tr>
<td>• Demonstrate cloud frameworks and technologies</td>
</tr>
<tr>
<td>• Define data intensive computing</td>
</tr>
<tr>
<td>• Demonstrate cloud applications</td>
</tr>
<tr>
<td><strong>Question paper pattern:</strong> The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.</td>
</tr>
<tr>
<td><strong>Text Books:</strong></td>
</tr>
<tr>
<td><strong>Reference Books:</strong></td>
</tr>
<tr>
<td>NIL</td>
</tr>
</tbody>
</table>
COMPUTER NETWORK LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

SEMESTER – V

Subject Code 17CSL57
IA Marks 40
Number of Lecture Hours/Week 01I + 02P
Exam Marks 60
Total Number of Lecture Hours 40
Exam Hours 03

CREDITS – 02

Description (If any):
For the experiments below modify the topology and parameters set for the experiment and take multiple rounds of reading and analyze the results available in log files. Plot necessary graphs and conclude. Use NS2/NS3.

Lab Experiments:

PART A

1. Implement 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.
2. Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.
3. Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.
4. Implement simple ESS and with transmitting nodes in wire-less LAN by simulation and determine the performance with respect to transmission of packets.
5. Implement and study the performance of GSM on NS2/NS3 (Using MAC layer) or equivalent environment.
6. Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.

PART B

Implement the following in Java:

7. Write a program for error detecting code using CRC-CCITT (16- bits).
8. Write a program to find the shortest path between vertices using bellman-ford algorithm.
9. Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.
10. Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.
11. Write a program for simple RSA algorithm to encrypt and decrypt the data.
12. Write a program for congestion control using leaky bucket algorithm.

Study Experiment / Project:
NIL

Course outcomes: The students should be able to:

• Analyze and Compare various networking protocols.
• Demonstrate the working of different concepts of networking.
• Implement and analyze networking protocols in NS2 / NS3

Conduction of Practical Examination:
1. All laboratory experiments are to be included for practical examination.
2. Students are allowed to pick one experiment from part A and part B with lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution: Procedure + Conduction + Viva: 100

<table>
<thead>
<tr>
<th>Part</th>
<th>8+35+7</th>
<th>=50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.
DBMS LABORATORY WITH MINI PROJECT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

SEMESTER – V

Subject Code 17CSL58 IA Marks 40
Number of Lecture Hours/Week 01I + 02P Exam Marks 60
Total Number of Lecture Hours 40 Exam Hours 03

CREDITS – 02

Description (If any):

PART-A: SQL Programming (Max. Exam Mks. 50)
- Design, develop, and implement the specified queries for the following problems using Oracle, MySQL, MS SQL Server, or any other DBMS under LINUX/Windows environment.
- Create Schema and insert at least 5 records for each table. Add appropriate database constraints.

PART-B: Mini Project (Max. Exam Mks. 30)
- Use Java, C#, PHP, Python, or any other similar front-end tool. All applications must be demonstrated on desktop/laptop as a stand-alone or web based application (Mobile apps on Android/IOS are not permitted.)

Lab Experiments:

Part A: SQL Programming

1. Consider the following schema for a Library Database:
   BOOK(Book_id, Title, Publisher_Name, Pub_Year)
   BOOK_AUTHORS(Book_id, Author_Name)
   PUBLISHER(Name, Address, Phone)
   BOOK_COPIES(Book_id, Branch_id, No-of_Copies)
   BOOK_LENDING(Book_id, Branch_id, Card_No, Date_Out, Due_Date)
   LIBRARY_BRANCH(Branch_id, Branch_Name, Address)

   Write SQL queries to
   1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.
   2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.
   3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
   4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
   5. Create a view of all books and its number of copies that are currently available in the Library.

2. Consider the following schema for Order Database:
   SALESMAN(Salesman_id, Name, City, Commission)
   CUSTOMER(Customer_id, Cust_Name, City, Grade, Salesman_id)
   ORDERS(Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)

   Write SQL queries to
   1. Count the customers with grades above Bangalore’s average.
   2. Find the name and numbers of all salesman who had more than one customer.
   3. List all the salesman and indicate those who have and don’t have customers in their cities (Use UNION operation.)
   4. Create a view that finds the salesman who has the customer with the highest order of a day.
5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

### 3
Consider the schema for Movie Database:
- **ACTOR(Act_id, Act_Name, Act_Gender)**
- **DIRECTOR(Dir_id, Dir_Name, Dir_Phone)**
- **MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)**
- **MOVIE_CAST(Act_id, Mov_id, Role)**
- **RATING(Mov_id, Rev_Stars)**

Write SQL queries to
1. List the titles of all movies directed by ‘Hitchcock’.
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
5. Update rating of all movies directed by ‘Steven Spielberg’ to 5.

### 4
Consider the schema for College Database:
- **STUDENT(USN, SName, Address, Phone, Gender)**
- **SEMSEC(SSID, Sem, Sec)**
- **CLASS(USN, SSID)**
- **SUBJECT(Subcode, Title, Sem, Credits)**
- **IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)**

Write SQL queries to
1. List all the student details studying in fourth semester ‘C’ section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN ‘1BI17CS101’ in all subjects.
4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:
   - If FinalIA = 17 to 20 then CAT = ‘Outstanding’
   - If FinalIA = 12 to 16 then CAT = ‘Average’
   - If FinalIA < 12 then CAT = ‘Weak’
   - Give these details only for 8th semester A, B, and C section students.

### 5
Consider the schema for Company Database:
- **EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo)**
- **DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate)**
- **DLOCATION(DNo, DLoc)**
- **PROJECT(PNo, PName, PLocation, DNo)**
- **WORKS_ON(SSN, PNo, Hours)**

Write SQL queries to
1. Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that controls the project.
2. Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.
3. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department.
4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).
5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

Part B: Mini project
- For any problem selected, write the ER Diagram, apply ER-mapping rules, normalize the relations, and follow the application development process.
- Make sure that the application should have five or more tables, at least one trigger and one stored procedure, using suitable frontend tool.
- Indicative areas include; health care, education, industry, transport, supply chain, etc.

Course outcomes: The students should be able to:
- Use Structured Query Language (SQL) for database Creation and manipulation.
- Demonstrate the working of different concepts of DBMS
- Implement and test the project developed for an application.

Conduction of Practical Examination:
1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 40 Marks.
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
   a) Part A: Procedure + Conduction + Viva: **09 + 42 +09 =60 Marks**
7. Part B: Demonstration + Report + Viva voce = **20+14+06 = 40 Marks**
8. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.
# CRYPTOGRAPHY, NETWORK SECURITY AND CYBER LAW

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2017 - 2018)

**SEMESTER – VI**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS61</td>
<td></td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS61</td>
<td>40</td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

## Module – 1


**Teaching Hours:** 10 Hours

## Module – 2


**Teaching Hours:** 10 Hours

## Module – 3


**Teaching Hours:** 10 Hours

## Module – 4


**Teaching Hours:** 10 Hours

## Module – 5

IT act aim and objectives, Scope of the act, Major Concepts, Important provisions, Attribution, acknowledgement, and dispatch of electronic records, Secure electronic records and secure digital signatures, Regulation of certifying authorities: Appointment of Controller and Other officers, Digital Signature certificates, Duties of Subscribers, Penalties and adjudication, The cyber regulations appellate tribunal, Offences, Network service providers not to be liable in certain cases, Miscellaneous Provisions.

**Teaching Hours:** 10 Hours

### Course outcomes:
The students should be able to:

- Discuss the cryptography and its need to various applications
- Design and Develop simple cryptography algorithms
• Understand the cyber security and need cyber Law

**Question paper pattern:**
The question paper will have TEN questions.  
There will be TWO questions from each module.  
Each question will have questions covering all the topics under a module.  
The students will have to answer FIVE full questions, selecting ONE full question from each module.

<table>
<thead>
<tr>
<th>Text Books</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Reference Books</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Cyber security and Cyber Laws, Alfred Basta, Nadine Basta, Mary brown, ravinndrakumar, Cengage learning</td>
</tr>
</tbody>
</table>
## COMPUTER GRAPHICS AND VISUALIZATION

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

### SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS62</td>
<td>40</td>
<td>60</td>
<td>50</td>
<td>3</td>
</tr>
</tbody>
</table>

### CREDITS – 04

### Module – 1


**Teaching Hours:** 10 Hours

### Text-1: Chapter -1: 1-1 to 1-9,2-1 to 2-9 (Excluding 2-5),3-1 to 3-5,3-9,3-20

### Module – 2

**Overview:** Fill area Primitives, 2D Geometric Transformations and 2D viewing: Fill area Primitives: Polygon fill-areas, OpenGL polygon fill area functions, fill area attributes, general scan line polygon fill algorithm, OpenGL fill-area attribute functions. 2D Geometric Transformations: Basic 2D Geometric Transformations, matrix representations and homogeneous coordinates. Inverse transformations, 2D Composite transformations, other 2D transformations, raster methods for geometric transformations, OpenGL raster transformations, OpenGL geometric transformations function, 2D viewing: 2D viewing pipeline, OpenGL 2D viewing functions.

**Teaching Hours:** 10 Hours

### Text-1: Chapter 3-14 to 3-16,4-9,4-10,4-14,5-1 to 5-7,5-17,6-1,6-4

### Module – 3

**Overview:** Clipping, 3D Geometric Transformations, Color and Illumination Models: Clipping: clipping window, normalization and viewport transformations, clipping algorithms, 2D point clipping, 2D line clipping algorithms: cohen-sutherland line clipping only -polygon fill area clipping; Sutherland-Hodgeman polygon clipping algorithm only. 3D Geometric Transformations: 3D translation, rotation, scaling, composite 3D transformations, other 3D transformations, affine transformations, OpenGL geometric transformations functions. Color Models: Properties of light, color models, RGB and CMY color models. Illumination Models: Light sources, basic illumination models-Ambient light, diffuse reflection, specular and phong model, Corresponding openGL functions.

**Teaching Hours:** 10 Hours

### Text-1: Chapter :6-2 to 6-08 (Excluding 6-4),5-9 to 5-17 (Excluding 5-15),12-1,12-2,12-4,12-6,10-1,10-3

### Module – 4

**Overview:** 3D Viewing and Visible Surface Detection: 3D Viewing: 3D viewing concepts, 3D viewing pipeline, 3D viewing coordinate parameters, Transformation from

**Teaching Hours:** 10 Hours

Text-1: Chapter: 7-1 to 7-10 (Excluding 7-7), 9-1 to 9-3, 9-14

Module – 5

Input & interaction, Curves and Computer Animation: Input and Interaction:
Input devices, clients and servers, Display Lists, Display Lists and Modelling, Programming Event Driven Input, Menus Picking, Building Interactive Models, Animating Interactive programs, Design of Interactive programs, Logic operations. Curved surfaces, quadric surfaces, OpenGL Quadric-Surface and Cubic-Surface Functions, Bezier Spline Curves, Bezier surfaces, OpenGL curve functions. Corresponding openGL functions.

Text-1: Chapter: 8-3 to 8-6 (Excluding 8-5), 8-9, 8-10, 8-11, 3-8, 8-18, 13-11, 3-2, 13-3, 13-4, 13-10

Text-2: Chapter 3: 3-1 to 3.11: Input & interaction

Course outcomes: The students should be able to:

- Design and implement algorithms for 2D graphics primitives and attributes.
- Illustrate Geometric transformations on both 2D and 3D objects.
- Understand the concepts of clipping and visible surface detection in 2D and 3D viewing, and Illumination Models.
- Discuss about suitable hardware and software for developing graphics packages using OpenGL.

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:


Reference Books:

1. James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes Computer graphics with OpenGL: pearson education
## SYSTEM SOFTWARE AND COMPILER DESIGN

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

### SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Total Number of Lecture Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS63</td>
<td>40</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>60</td>
<td>03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CREDITS – 04</th>
</tr>
</thead>
</table>

### Module – 1

Introduction to System Software, Machine Architecture of SIC and SIC/XE.

**Assemblers:** Basic assembler functions, machine dependent assembler features, machine independent assembler features, assembler design options.

**Macroprocessors:** Basic macro processor functions,

**Text book 1:** Chapter 1: 1.1,1.2,1.3.1,1.3.2, Chapter2 : 2.1-2.4,Chapter4: 4.1.1,4.1.2

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

### Module – 2

**Loaders and Linkers:** Basic Loader Functions, Machine Dependent Loader Features, Machine Independent Loader Features, Loader Design Options, Implementation Examples.

**Text book 1:** Chapter 3 ,3.1 -3.5

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

### Module – 3

**Introduction:** Language Processors, The structure of a compiler, The evaluation of programming languages, The science of building compiler, Applications of compiler technology, Programming language basics

**Lexical Analysis:** The role of lexical analyzer, Input buffering, Specifications of token, recognition of tokens, lexical analyzer generator, Finite automate.

**Text book 2:** Chapter 1 1.1-1.6 Chapter 3 3.1 – 3.6

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

### Module – 4

**Syntax Analysis:** Introduction, Role Of Parsers, Context Free Grammars, Writing a grammar, Top Down Parsers, Bottom-Up Parsers, Operator-Precedence Parsing

**Text book 2:** Chapter 4 4.1 4.2 4.3 4.4 4.5 4.6  Text book 1 : 5.1.3

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

### Module – 5

**Syntax Directed Translation, Intermediate code generation, Code generation**

**Text book 2:** Chapter 5.1, 5.2, 5.3, 6.1, 6.2, 8.1, 8.2

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Hours</td>
</tr>
</tbody>
</table>

**Course outcomes:**
- Illustrate system software such as assemblers, loaders, linkers and macroprocessors
- Design and develop lexical analyzers, parsers and code generators
- Discuss about lex and yacc tools for implementing different concepts of system software

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

<table>
<thead>
<tr>
<th>Reference Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. System programming and Compiler Design, K C Louden, Cengage Learning</td>
</tr>
<tr>
<td>3. System software and operating system by D. M. Dhamdhere TMG</td>
</tr>
</tbody>
</table>
**OPERATING SYSTEMS**  
[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2017 - 2018)  
SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS64</td>
<td>40</td>
<td>60</td>
<td>50</td>
</tr>
</tbody>
</table>

**CREDITS – 04**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot. Process Management Process concept; Process scheduling; Operations on processes; Inter process communication</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. Process Synchronization: Synchronization: The critical section problem; Peterson’s solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deadlocks : Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. Memory Management: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. File System, Implementation of File System: File system: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secondary Storage Structures, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. Case Study: The Linux Operating System: Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output;</td>
<td>10 Hours</td>
</tr>
</tbody>
</table>
Inter-process communication.

**Course outcomes:** The students should be able to:

- Demonstrate need for OS and different types of OS
- Discuss suitable techniques for management of different resources
- Illustrate processor, memory, storage and file system commands
- Explain the different concepts of OS in platform of usage through case studies

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books**
DATA MINING AND DATA WAREHOUSING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS651</td>
<td></td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

CREDITS – 03

| Module – 1 | Data Warehousing & modeling: | Basic Concepts: Data Warehousing: A multitier Architecture, Data warehouse models: Enterprise warehouse, Data mart and virtual warehouse, Extraction, Transformation and loading, Data Cube: A multidimensional data model, Stars, Snowflakes and Fact constellations: Schemas for multidimensional Data models, Dimensions: The role of concept Hierarchies, Measures: Their Categorization and computation, Typical OLAP Operations. | 8 Hours |
| Module – 2 | Data warehouse implementation & Data mining: | Efficient Data Cube computation: An overview, Indexing OLAP Data: Bitmap index and join index, Efficient processing of OLAP Queries, OLAP server Architecture ROLAP versus MOLAP Versus HOLAP.: Introduction: What is data mining, Challenges, Data Mining Tasks, Data: Types of Data, Data Quality, Data Preprocessing, Measures of Similarity and Dissimilarity. | 8 Hours |
| Module – 3 | Association Analysis: | Association Analysis: Problem Definition, Frequent Item set Generation, Rule generation. Alternative Methods for Generating Frequent Item sets, FP-Growth Algorithm, Evaluation of Association Patterns. | 8 Hours |
| Module – 4 | Classification : | Decision Trees Induction, Method for Comparing Classifiers, Rule Based Classifiers, Nearest Neighbor Classifiers, Bayesian Classifiers. | 8 Hours |
| Module – 5 | Clustering Analysis: | Overview, K-Means, Agglomerative Hierarchical Clustering, DBSCAN, Cluster Evaluation, Density-Based Clustering, Graph-Based Clustering, Scalable Clustering Algorithms. | 8 Hours |

Course outcomes: The students should be able to:
- Understands data mining problems and implement the data warehouse
- Demonstrate the association rules for a given data pattern.
- Discuss between classification and clustering solution.

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:
1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining,
<table>
<thead>
<tr>
<th></th>
<th>Pearson, First impression, 2014.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>Jiawei Han, Micheline Kamber, Jian Pei: Data Mining - Concepts and Techniques, 3rd Edition, Morgan Kaufmann Publisher, 2012.</td>
</tr>
</tbody>
</table>

**Reference Books:**

## SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

*As per Choice Based Credit System (CBCS) scheme*  
*(Effective from the academic year 2017 - 2018)*

### SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS652</td>
<td>40</td>
<td>3</td>
<td>60</td>
<td>40</td>
<td>3</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

### Module – 1

**Introduction**: what is a design pattern? describing design patterns, the catalog of design pattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern. What is object-oriented development? , key concepts of object oriented design other related concepts, benefits and drawbacks of the paradigm 8 Hours

### Module – 2

**Analysis a System**: overview of the analysis phase, stage 1: gathering the requirements functional requirements specification, defining conceptual classes and relationships, using the knowledge of the domain. Design and Implementation, discussions and further reading. 8 Hours

### Module – 3

**Design Pattern Catalog**: Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy. 8 Hours

### Module – 4

**Interactive systems and the MVC architecture**: Introduction, The MVC architectural pattern, analyzing a simple drawing program, designing the system, designing of the subsystems, getting into implementation, implementing undo operation, drawing incomplete items, adding a new feature, pattern based solutions. 8 Hours

### Module – 5

**Designing with Distributed Objects**: Client server system, java remote method invocation, implementing an object oriented system on the web (discussions and further reading) a note on input and output, selection statements, loops arrays. 8 Hours

### Course outcomes:

- Design and implement codes with higher performance and lower complexity
- Demonstrate code qualities needed to keep code flexible
- Illustrate design principles and be able to assess the quality of a design with respect to these principles.
- Explain principles in the design of object oriented systems.
- Understand a range of design patterns.
- Discuss suitable patterns in specific contexts

### Question paper pattern:

The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

### Text Books:
1. **Object-oriented analysis, design and implementation**, Brahma Dathan, Sarnathrammath, Universities press, 2013

**Reference Books:**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Frank Bachmann, Regine Meunier, Hans Rohnert “Pattern Oriented Software Architecture” – Volume 1, 1996.</td>
</tr>
<tr>
<td>Module – 1</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Introduction, Linear Programming:</strong> Introduction: The origin, nature and impact of OR; Defining the problem and gathering data; Formulating a mathematical model; Deriving solutions from the model; Testing the model; Preparing to apply the model; Implementation.</td>
<td></td>
</tr>
<tr>
<td><strong>Introduction to Linear Programming Problem (LPP):</strong> Prototype example, Assumptions of LPP, Formulation of LPP and Graphical method various examples.</td>
<td></td>
</tr>
<tr>
<td>Teaching Hours</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simplex Method – 1:</strong> The essence of the simplex method; Setting up the simplex method; Types of variables, Algebra of the simplex method; the simplex method in tabular form; Tie breaking in the simplex method, Big M method, Two phase method.</td>
</tr>
<tr>
<td><strong>Simplex Method – 2: Duality Theory</strong> - The essence of duality theory, Primal dual relationship, conversion of primal to dual problem and vice versa. The dual simplex method.</td>
</tr>
<tr>
<td>Teaching Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Game Theory:</strong> Game Theory: The formulation of two persons, zero sum games; saddle point, maximin and minimax principle, Solving simple games-a prototype example; Games with mixed strategies; Graphical solution procedure.</td>
</tr>
<tr>
<td><strong>Metaheuristics:</strong> The nature of Metaheuristics, Tabu Search, Simulated Annealing, Genetic Algorithms.</td>
</tr>
<tr>
<td>Teaching Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course outcomes: The students should be able to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Explain optimization techniques for various problems.</td>
</tr>
<tr>
<td>• Understand the given problem as transportation and assignment problem and solve.</td>
</tr>
<tr>
<td>• Illustrate game theory for decision support system.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question paper pattern:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The question paper will have TEN questions.</td>
</tr>
<tr>
<td>There will be TWO questions from each module.</td>
</tr>
<tr>
<td>Each question will have questions covering all the topics under a module.</td>
</tr>
<tr>
<td>The students will have to answer FIVE full questions, selecting ONE full question from each module.</td>
</tr>
</tbody>
</table>
**Text Books:**


**Reference Books:**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Characterization of Distributed Systems: Introduction, Examples of DS, Resource sharing and the Web, Challenges</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Models: Architectural Models, Fundamental Models</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module – 2</td>
<td>Inter Process Communication: Introduction, API for Internet Protocols, External Data Representation and Marshalling, Client – Server Communication, Group Communication</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Distributed Objects and RMI: Introduction, Communication between Distributed Objects, RPC, Events and Notifications</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module – 3</td>
<td>Operating System Support: Introduction, The OS layer, Protection, Processes and Threads, Communication and Invocation, Operating system architecture</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Distributed File Systems: Introduction, File Service architecture, Sun Network File System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module – 4</td>
<td>Time and Global States: Introduction, Clocks, events and process status, Synchronizing physical clocks, Logical time and logical clocks, Global states</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Coordination and Agreement: Introduction, Distributed mutual exclusion, Elections</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module – 5</td>
<td>Distributed Transactions: Introduction, Flat and nested distributed transactions, Atomic commit protocols, Concurrency control in distributed transactions, distributed deadlocks</td>
<td>8 Hours</td>
</tr>
<tr>
<td>Course outcomes: The students should be able to:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Explain the characteristics of a distributed system along with its and design challenges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Illustrate the mechanism of IPC between distributed objects</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Describe the distributed file service architecture and the important characteristics of SUN NFS.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Discuss concurrency control algorithms applied in distributed transactions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question paper pattern:</td>
<td>The question paper will have TEN questions.</td>
<td></td>
</tr>
<tr>
<td>There will be TWO questions from each module.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Each question will have questions covering all the topics under a module.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The students will have to answer FIVE full questions, selecting ONE full question from each module.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference Books:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# MOBILE APPLICATION DEVELOPMENT

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)

**SEMESTER – VI**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS661</td>
<td>40</td>
<td>3</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

### Module – 1
Get started, Build your first app, Activities, Testing, debugging and using support libraries

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Module – 2
User Interaction, Delightful user experience, Testing your UI

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Module – 3
Background Tasks, Triggering, scheduling and optimizing background tasks

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Module – 4
All about data, Preferences and Settings, Storing data using SQLite, Sharing data with content providers, Loading data using Loaders

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Module – 5
Permissions, Performance and Security, Firebase and AdMob, Publish

<table>
<thead>
<tr>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Hours</td>
</tr>
</tbody>
</table>

### Course outcomes:
The students should be able to:

- Design and Develop Android application by setting up Android development environment
- Implement adaptive, responsive user interfaces that work across a wide range of devices.
- Explain long running tasks and background work in Android applications
- Demonstrate methods in storing, sharing and retrieving data in Android applications
- Discuss the performance of android applications and understand the role of permissions and security
- Describe the steps involved in publishing Android application to share with the world

### Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

### Text Books:

### Reference Books:
## BIG DATA ANALYTICS

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2017-2018)

### SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS662</td>
<td>40</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

### CREDITS – 03

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction to Data Analytics and Decision Making</strong>:</td>
<td>08 Hours</td>
</tr>
<tr>
<td><strong>Describing the Distribution of a Single Variable</strong>:</td>
<td></td>
</tr>
<tr>
<td><strong>Finding Relationships among Variables</strong>:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Probability and Probability Distributions</strong>:</td>
<td>08 Hours</td>
</tr>
<tr>
<td><strong>Normal, Binormal, Poisson, and Exponential Distributions</strong>:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decision Making under Uncertainty</strong>:</td>
<td>08 Hours</td>
</tr>
</tbody>
</table>
### Module – 4

**Confidence Interval Estimation:** Introduction, Sampling Distributions, The t Distribution, Other Sampling Distributions, Confidence Interval for a Mean, Confidence Interval for a Total, Confidence Interval for a Proportion, Confidence Interval for a Standard Deviation, Confidence Interval for the Difference between Means, Independent Samples, Paired Samples, Confidence Interval for the Difference between Proportions, Sample Size Selection, Sample Size Selection for Estimation of the Mean, Sample Size Selection for Estimation of Other Parameters.


### Module – 5


**Regression Analysis:** Statistical Inference: Introduction, The Statistical Model, Inferences About the Regression Coefficients, Sampling Distribution of the Regression Coefficients, Hypothesis Tests for the Regression Coefficients and p-Values, A Test for the Overall Fit: The ANOVA Table, Multicollinearity, Include/Exclude Decisions, Stepwise Regression, Outliers, Violations of Regression Assumptions, Nonconstant Error Variance, Nonnormality of Residuals, Autocorrelated Residuals, Prediction.

### Course outcomes:
The students should be able to:
- Explain the importance of data and data analysis
- Interpret the probabilistic models for data
- Illustrate hypothesis, uncertainty principle
- Demonstrate the regression analysis

### Question paper pattern:
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

<table>
<thead>
<tr>
<th>Text Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. S C Albright and W L Winston, Business analytics: data analysis and decision making, 5/e Cenage Learning</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
WIRELESS NETWORKS AND MOBILE COMPUTING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

SEMESTER – VI

Subject Code 17CS663 IA Marks 40
Number of Lecture Hours/Week 3 Exam Marks 60
Total Number of Lecture Hours 40 Exam Hours 03

CREDITS – 03

Module – 1
Mobile Communication, Mobile Computing, Mobile Computing Architecture,
Mobile Devices Mobile System Networks, Data Dissemination, Mobility
Management, Security Cellular Networks and Frequency Reuse, Mobile
Smartphone, Smart Mobiles, and Systems Handheld Pocket Computers,
Handheld Devices, Smart Systems, Limitations of Mobile Devices
Automotive Systems

8 Hours

Module – 2
GSM-Services and System Architecture, Radio Interfaces of GSM, Protocols of
GSM Localization, Call Handling Handover, Security, New Data Services,
General Packet Radio Service High-speed Circuit Switched Data, DECT,
Modulation, Multiplexing, Controlling the Medium Access Spread Spectrum,
Frequency Hopping Spread Spectrum (FHSS),Coding Methods, Code Division
Multiple Access, IMT-2000 3G Wireless Communication Standards, WCDMA
3G Communications Standards,CDMA2000 3G Communication Standards,
I-mode, OFDM, High Speed Packet Access (HSPA) 3G Network
Long-term Evolution, WiMaxRel 1.0 IEEE 802.16e, Broadband Wireless
Access,4G Networks, Mobile Satellite Communication Networks

8 Hours

Module – 3
IP and Mobile IP Network Layers, Packet Delivery and Handover Management
Location Management, Registration, Tunnelling and Encapsulation, Route
Optimization Dynamic Host Configuration Protocol, VoIP, IPsec
Conventional TCP/IP Transport Layer Protocols, Indirect TCP, Snooping TCP
Mobile TCP, Other Methods of Mobile TCP-layer Transmission ,TCP over
2.5G/3G Mobile Networks

8 Hours

Module – 4
Data Organization, Database Transactional Models – ACID Rules, Query
Processing Data Recovery Process, Database Hoarding Techniques , Data
Caching, Client-Server Computing for Mobile Computing and Adaptation
Adaptation Software for Mobile Computing, Power-Aware Mobile Computing,
Context-aware Mobile Computing

8 Hours

Module – 5
Communication Asymmetry, Classification of Data-delivery Mechanisms, Data
Dissemination Broadcast Models, Selective Tuning and Indexing techniques,
Digital Audio Broadcasting (DAB), Digital Video Broadcasting
Synchronization, Synchronization Software for Mobile Devices, Synchronization
Software for Mobile Devices
SyncML-Synchronization Language for Mobile Computing,Sync4J (Funambol),
Synchronized Multimedia Markup Language (SMIL)

8 Hours

Course outcomes: The students should be able to:
- Understand the various mobile communication systems.
- Describe various multiplexing systems used in mobile computing.
- Explain the use and importance of data synchronization in mobile computing

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
PYTHON APPLICATION PROGRAMMING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)
SEMESTER – VI

Subject Code  17CS664
IA Marks  40
Number of Lecture Hours/Week  3
Exam Marks  60
Total Number of Lecture Hours  40
Exam Hours  03

CREDITS – 03

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why should you learn to write programs, Variables, expressions and statements, Conditional execution, Functions</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration, Strings, Files</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lists, Dictionaries, Tuples, Regular Expressions</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes and objects, Classes and functions, Classes and methods</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>8 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Networked programs, Using Web Services, Using databases and SQL</td>
<td></td>
</tr>
</tbody>
</table>

Course outcomes: The students should be able to:

- Understand Python syntax and semantics and be fluent in the use of Python flow control and functions.
- Demonstrate proficiency in handling Strings and File Systems.
- Implement Python Programs using core data structures like Lists, Dictionaries and use Regular Expressions.
- Interpret the concepts of Object-Oriented Programming as used in Python.
- Implement exemplary applications related to Network Programming, Web Services and Databases in Python.

Question paper pattern:
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

Text Books:

Reference Books:
### SERVICE ORIENTED ARCHITECTURE

[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2017 -2018)

#### SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS665</td>
<td>40</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

#### CREDITS – 03

<table>
<thead>
<tr>
<th>Module – 1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>**SOA BASICS:**Software Architecture; Need for Software Architecture, Objectives of Software Architecture, Types of IT Architecture, Architecture Patterns and Styles, <strong>Service oriented Architecture:</strong> Service Orientation in Daily Life, Evolution of SOA, Drives for SOA, Dimension of SOA, Key components, perspective of SOA, <strong>Enterprise-wide SOA:</strong> Considerations for Enterprise-Wide SOA, Strawman Architecture For Enterprise-Wide-SOA-Enterprise, SOA-Layers, Application Development Process, SOA Methodology For Enterprise</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

**Text 1:** Ch2: 2.1 – 2.4; Ch3: 3.1-3.7; Ch4: 4.1 – 4.5

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enterprise Applications:</strong> Architecture Considerations, Solution Architecture for enterprise application, <strong>Software platforms for enterprise Applications:</strong> Package Application Platforms, Enterprise Application Platforms, <strong>Service-oriented-Enterprise Applications:</strong> Considerations for Service-Oriented Enterprise Applications, Patterns for SOA, Pattern-Based Architecture for Service-Oriented Enterprise Application (java reference model only). Composite Applications, SOA programming models.</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

**Text 1:** Ch5: 5.1, 5.2, 6.1, 6.2 (PageNo 74-81), 7.1 – 7.5

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SOA ANALYSIS AND DESIGN:</strong> Need For Models, Principles of Service Design, Design of Activity Services, Design of Datasets, Design of Client services and Design of business process services, <strong>Technologies of SOA:</strong> Technologies For Service Enablement, Technologies For Service Integration, Technologies for Service orchestration.</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

**Text 1:** Ch 8: 8.1 – 8.6, 9.1 – 9.3

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business case for SOA:</strong> Stakeholder OBJECTIVES, Benefits of SOA, Cost Savings, Return on Investment, SOA Governance, <strong>Security and implementation:</strong> SOA Governance, SOA Security, approach for enterprise wide SOA implementation, <strong>Trends in SOA:</strong> Technologies in Relation to SOA, Advances in SOA.</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

**Text 1:** Ch 10: 10.1 -10.4, Ch 11: 11.1 to 11.3, Ch12: 12.2, 12.3

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SOA Technologies-PoC:</strong> Loan Management System (LMS), PoC-Requirements Architectures of LMS <strong>SOA based integration:</strong> integrating existing application, <strong>SOA best practices,</strong> Basic SOA using REST. Role of WSDL, SOAP and JAVA/XML Mapping in SOA.</td>
<td>8 Hours</td>
</tr>
</tbody>
</table>

**Text 1:** Page No 245-248; **ReferenceBook:** Chapter3; **Text 1:** Page No 307-310 **Text 2:** Ch 3, Ch4

### Course outcomes:
The students should be able to:
- Understand the different IT architectures
- Explain SOA based applications
- Illustrate web service and realization of SOA
- Discuss RESTful services

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
MULTI-CORE ARCHITECTURE AND PROGRAMMING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)
SEMESTER – VI

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS666</td>
<td>40</td>
<td>3</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

CREDITS – 03

Module – 1


Module – 2


Module – 3

Threading APIs: Threading APIs for Microsoft Windows, Win32/MFC Thread APIs, Threading APIs for Microsoft. NET Framework, Creating Threads, Managing Threads, Thread Pools, Thread Synchronization, POSIX Threads, Creating Threads, Managing Threads, Thread Synchronization, Signaling, Compilation and Linking.

Module – 4


Module – 5


**Course outcomes:** The students should be able to:

- Identify the issues involved in multicore architectures
- Explain fundamental concepts of parallel programming and its design issues
- Solve the issues related to multiprocessing and suggest solutions
- Discuss salient features of different multicore architectures and how they exploit parallelism
- Illustrate OpenMP and programming concept

**Question paper pattern:**
The question paper will have TEN questions.
There will be TWO questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer FIVE full questions, selecting ONE full question from each module.

**Text Books:**

**Reference Books:**
NIL
# SYSTEM SOFTWARE AND OPERATING SYSTEM LABORATORY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

**SEMESTER – VI**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CSL67</td>
<td></td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 02**

**Description (If any):**

Exercises to be prepared with minimum three files (Where ever necessary):

i. Header file.
ii. Implementation file.
iii. Application file where main function will be present.

The idea behind using three files is to differentiate between the developer and user sides. In the developer side, all the three files could be made visible. For the user side only header file and application files could be made visible, which means that the object code of the implementation file could be given to the user along with the interface given in the header file, hiding the source file, if required. Avoid I/O operations (printf/scanf) and use data input file where ever it is possible.

**Lab Experiments:**

1. a) Write a LEX program to recognize valid arithmetic expression. Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately.

   b) Write YACC program to evaluate arithmetic expression involving operators: +, -, *, and /.

2. Develop, Implement and Execute a program using YACC tool to recognize all strings ending with b preceded by na’s using the grammar a^n b (note: input n value).

3. Design, develop and implement YACC/C program to construct Predictive / LL(1) Parsing Table for the grammar rules: A →aBa, B →bB | ε. Use this table to parse the sentence: abba$

4. Design, develop and implement YACC/C program to demonstrate Shift Reduce Parsing technique for the grammar rules: E →E+T | T, T →T*F | F, F →(E) | id and parse the sentence: id + id * id.

5. Design, develop and implement a C/Java program to generate the machine code using Triples for the statement A = -B * (C +D) whose intermediate code in three-address form:

   \[
   \begin{align*}
   T1 &= -B \\
   T2 &= C + D \\
   T3 &= T1 + T2 \\
   A &= T3
   \end{align*}
   \]

6. a) Write a LEX program to eliminate comment lines in a C program and copy the
resulting program into a separate file.
b) Write YACC program to recognize valid identifiers, operators and keywords in the
given text (C program) file.

7. Design, develop and implement a C/C++/Java program to simulate the working of
shortest remaining time and Round Robin (RR) scheduling algorithms. Experiment
with different quantum sizes for RR algorithm.

8. Design, develop and implement a C/C++/Java program to implement Banker’s
algorithm. Assume suitable input required to demonstrate the results.

9. Design, develop and implement a C/C++/Java program to implement page
replacement algorithms LRU and FIFO. Assume suitable input required to
demonstrate the results.

Study Experiment / Project:

Course outcomes: The students should be able to:

- Implement and demonstrate Lexer’s and Parser’s
- Implement different algorithms required for management, scheduling, allocation and
  communication used in operating system.

Conduction of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Students are allowed to pick one experiment from the lot.
- Strictly follow the instructions as printed on the cover page of answer script
- Marks distribution: Procedure + Conduction + Viva: 15 + 70 + 15 (100)
- Change of experiment is allowed only once and marks allotted to the procedure
  part to be made zero
### Lab Experiments:

**PART A**

Design, develop, and implement the following programs using OpenGL API

1. Implement Brenham’s line drawing algorithm for all types of slope.
   
   **Refer:** Text-1: Chapter 3.5
   
   **Refer:** Text-2: Chapter 8

2. Create and rotate a triangle about the origin and a fixed point.
   
   **Refer:** Text-1: Chapter 5-4

3. Draw a colour cube and spin it using OpenGL transformation matrices.
   
   **Refer:** Text-2: *Modelling a Coloured Cube*

4. Draw a color cube and allow the user to move the camera suitably to experiment with perspective viewing.
   
   **Refer:** Text-2: *Topic: Positioning of Camera*

5. Clip a lines using Cohen-Sutherland algorithm
   
   **Refer:** Text-1: Chapter 6.7
   
   **Refer:** Text-2: Chapter 8

6. To draw a simple shaded scene consisting of a tea pot on a table. Define suitably the position and properties of the light source along with the properties of the surfaces of the solid object used in the scene.
   
   **Refer:** Text-2: *Topic: Lighting and Shading*

7. Design, develop and implement recursively subdivide a tetrahedron to form 3D sierpinski gasket. The number of recursive steps is to be specified by the user.
   
   **Refer:** Text-2: *Topic: sierpinski gasket.*

8. Develop a menu driven program to animate a flag using Bezier Curve algorithm
   
   **Refer:** Text-1: Chapter 8-10

9. Develop a menu driven program to fill the polygon using scan line algorithm

### Project:

**PART B (MINI-PROJECT):**

Student should develop mini project on the topics mentioned below or similar applications using OpenGL API. Consider all types of attributes like color, thickness, styles, font, background, speed etc., while doing mini project.

(During the practical exam: the students should demonstrate and answer Viva-Voce)

**Sample Topics:**

Simulation of concepts of OS, Data structures, algorithms etc.

**Course outcomes:** The students should be able to:

- Apply the concepts of computer graphics
- Implement computer graphics applications using OpenGL
- Implement real world problems using OpenGL

**Conduction of Practical Examination:**
1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 40 Marks.
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
   a) Part A: Procedure + Conduction + Viva: 09 + 42 +09 =60 Marks
   b) Part B: Demonstration + Report + Viva voce = 20+14+06 = 40 Marks
7. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

**Reference books:**

WEB TECHNOLOGY AND ITS APPLICATIONS
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

SEMESTER – VII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>04</td>
<td>Exam Marks</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>50</td>
<td>Exam Hours</td>
</tr>
</tbody>
</table>

CREDITS – 04

Module – 1


Module – 2

HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility, Microformats, Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks.

Module – 3


Module – 4

PHP Arrays and Superglobals, Arrays, $_GET and $_POST Superglobal Arrays, $_SERVER Array, $_Files Array, Reading/Writing Files, PHP Classes and Objects, Object-Oriented Overview, Classes and Objects in PHP, Object Oriented Design, Error Handling and Validation, What are Errors and Exceptions?, PHP Error Reporting, PHP Error and Exception Handling

Module – 5


Course Outcomes: After studying this course, students will be able to

- Define HTML and CSS syntax and semantics to build web pages.
- Understand the concepts of Construct, visually format tables and forms using HTML using CSS
- Develop Client-Side Scripts using JavaScript and Server-Side Scripts using PHP to generate and display the contents dynamically.
- List the principles of object oriented development using PHP
- Illustrate JavaScript frameworks like jQuery and Backbone which facilitates
developer to focus on core features.

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**
ADVANCED COMPUTER ARCHITECTURES
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)
SEMESTER – VII

Subject Code 17CS72  IA Marks 40
Number of Lecture Hours/Week 4  Exam Marks 60
Total Number of Lecture Hours 50  Exam Hours 03

CREDITS – 04

Module – 1

Module – 2
Hardware Technologies: Processors and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors, Memory Hierarchy Technology, Virtual Memory Technology.

Module – 3

Module – 4

Module – 5
Software for parallel programming: Parallel Models, Languages, and Compilers ,Parallel Programming Models, Parallel Languages and Compilers ,Dependence Analysis of Data Arrays ,Parallel Program Development and Environments, Synchronization and Multiprocessing Modes, Instruction and System Level Parallelism, Instruction Level Parallelism ,Computer Architecture ,Contents, Basic Design Issues ,Problem Definition ,Model of a Typical Processor ,Compiler-detected Instruction Level Parallelism ,Operand Forwarding ,Reorder Buffer, Register Renaming ,Tomasulo’s Algorithm ,Branch Prediction, Limitations in Exploiting Instruction Level Parallelism ,Thread Level Parallelism.

Course outcomes: The students should be able to:
- Understand the concepts of parallel computing and hardware technologies
- Illustrate and contrast the parallel architectures
- Recall parallel programming concepts

**Question paper pattern**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**
### Module – 1

**Introduction:** Well posed learning problems, Designing a Learning system, Perspective and Issues in Machine Learning.

**Concept Learning:** Concept learning task, Concept learning as search, Find-S algorithm, Version space, Candidate Elimination algorithm, Inductive Bias.

**Text Book1, Sections:** 1.1 – 1.3, 2.1-2.5, 2.7

- **Teaching Hours:** 10

### Module – 2

**Decision Tree Learning:** Decision tree representation, Appropriate problems for decision tree learning, Basic decision tree learning algorithm, hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning.

**Text Book1, Sections:** 3.1-3.7

- **Teaching Hours:** 10

### Module – 3

**Artificial Neural Networks:** Introduction, Neural Network representation, Appropriate problems, Perceptrons, Backpropagation algorithm.

**Text book 1, Sections:** 4.1 – 4.6

- **Teaching Hours:** 8

### Module – 4

**Bayesian Learning:** Introduction, Bayes theorem, Bayes theorem and concept learning, ML and LS error hypothesis, ML for predicting probabilities, MDL principle, Naive Bayes classifier, Bayesian belief networks, EM algorithm

**Text book 1, Sections:** 6.1 – 6.6, 6.9, 6.11, 6.12

- **Teaching Hours:** 10

### Module – 5

**Evaluating Hypothesis:** Motivation, Estimating hypothesis accuracy, Basics of sampling theorem, General approach for deriving confidence intervals, Difference in error of two hypothesis, Comparing learning algorithms.

**Instance Based Learning:** Introduction, k-nearest neighbor learning, locally weighted regression, radial basis function, cased-based reasoning.

**Reinforcement Learning:** Introduction, Learning Task, Q Learning

**Text book 1, Sections:** 5.1-5.6, 8.1-8.5, 13.1-13.3

- **Teaching Hours:** 12

### Course Outcomes:

- After studying this course, students will be able to:
  - Recall the problems for machine learning. And select the either supervised, unsupervised or reinforcement learning.
  - Understand theory of probability and statistics related to machine learning
  - Illustrate concept learning, ANN, Bayes classifier, k nearest neighbor, Q

### Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

<table>
<thead>
<tr>
<th>Text Books:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Reference Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, h The Elements of Statistical Learning, 2nd edition, springer series in statistics.</td>
</tr>
<tr>
<td>2. Ethem Alpaydın, Introduction to machine learning, second edition, MIT press.</td>
</tr>
</tbody>
</table>
NATURAL LANGUAGE PROCESSING
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)
SEMESTER – VII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS741</td>
<td></td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>Number of Lecture Hours/Week</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>40</td>
</tr>
</tbody>
</table>

CREDITS – 03

Module – 1

Overview and language modeling:

Module – 2

Word level and syntactic analysis:

Module – 3

Extracting Relations from Text: From Word Sequences to Dependency Paths:
Introduction, Subsequence Kernels for Relation Extraction, A Dependency-Path Kernel for Relation Extraction and Experimental Evaluation.

Mining Diagnostic Text Reports by Learning to Annotate Knowledge Roles:
Introduction, Domain Knowledge and Knowledge Roles, Frame Semantics and Semantic Role Labeling, Learning to Annotate Cases with Knowledge Roles and Evaluations.


Module – 4


Automatic Document Separation: A Combination of Probabilistic Classification and Finite-State Sequence Modeling:
Introduction, Related Work, Data Preparation, Document Separation as a Sequence Mapping Problem, Results.

Evolving Explanatory Novel Patterns for Semantically-Based Text Mining:
Related Work, A Semantically Guided Model for Effective Text Mining.

Module – 5

INFORMATION RETRIEVAL AND LEXICAL RESOURCES:

Teaching Hours

8 Hours
**Course outcomes:** The students should be able to:

- Analyze the natural language text.
- Define the importance of natural language.
- Understand the concepts Text mining.
- Illustrate information retrieval techniques.

**Question paper pattern:**
The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

### CLOUD COMPUTING AND ITS APPLICATIONS

*As per Choice Based Credit System (CBCS) scheme*

(Effective from the academic year 2017 - 2018)

**SEMESTER – VII**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS742</td>
<td>40</td>
<td>60</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

#### Module – 1


Virtualization, Introduction, Characteristics of Virtualized, Environments Taxonomy of Virtualization Techniques, Execution Virtualization, Other Types of Virtualization, Virtualization and Cloud Computing, Pros and Cons of Virtualization, Technology Examples Xen: Paravirtualization, VMware: Full Virtualization, Microsoft Hyper-V

#### Module – 2


#### Module – 3


Parameter Sweep Applications, MPI Applications, Workflow Applications with Task Dependencies, Aneka Task-Based Programming, Task Programming Model, Developing Applications with the Task Model, Developing Parameter Sweep Application, Managing Workflows.

### Module – 4


| 8 Hours |

### Module – 5


| 8 Hours |

**Course outcomes:** The students should be able to:

- Understand the concepts of cloud computing, virtualization and classify services of cloud computing
- Illustrate architecture and programming in cloud
- Define the platforms for development of cloud applications and List the application of cloud.

**Question paper pattern:**

The question paper will have ten questions.

There will be 2 questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

INFORMATION AND NETWORK SECURITY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)
SEMESTER – VII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS743</td>
<td></td>
<td>3</td>
<td>60</td>
<td>40</td>
<td>3</td>
</tr>
</tbody>
</table>

CREDITS – 03

Module – 1

Module – 2.

Module – 3
Random number generation Providing freshness Fundamentals of entity authentication Passwords Dynamic password schemes Zero-knowledge mechanisms Further reading Cryptographic Protocols Protocol basics From objectives to a protocol Analysing a simple protocol Authentication and key establishment protocols

Module – 4
Key management fundamentals Key lengths and lifetimes Key generation Key establishment Key storage Key usage Governing key management Public-Key Management Certification of public keys The certificate lifecycle Public-key management models Alternative approaches

Module – 5
Cryptographic Applications Cryptography on the Internet Cryptography for wireless local area networks Cryptography for mobile telecommunications Cryptography for secure payment card transactions Cryptography for video broadcasting Cryptography for identity cards Cryptography for home users

Course outcomes: The students should be able to:

- Analyze the Digitalis security lapses
- Illustrate the need of key management

Question paper pattern:
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:
<table>
<thead>
<tr>
<th>Reference Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module – 1</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>Module – 2</td>
</tr>
<tr>
<td>Module – 3</td>
</tr>
<tr>
<td>Module – 4</td>
</tr>
<tr>
<td>Module – 5</td>
</tr>
</tbody>
</table>

**Course outcomes:** The students should be able to:
- Understand the working of Unix Systems
- Illustrate the application/service over a UNIX system.
**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

<table>
<thead>
<tr>
<th>Text Books:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Reference Books:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module – 1</td>
</tr>
<tr>
<td>------------</td>
</tr>
</tbody>
</table>
| Introduction to soft computing: ANN, FS, GA, SI, ES, Comparing among intelligent systems  
ANN: introduction, biological inspiration, BNN&ANN, classification, first Generation NN, perceptron, illustrative problems  
**Text Book 1: Chapter1: 1.1-1.8, Chapter2: 2.1-2.6** | 8 Hours |

<table>
<thead>
<tr>
<th>Module – 2</th>
<th>8 Hours</th>
</tr>
</thead>
</table>
| Adaline, Medaline, ANN: (2nd generation), introduction, BPN, KNN, HNN, BAM, RBF, SVM and illustrative problems  
**Text Book 1: Chapter2: 3.1, 3.2, 3.3, 3.6, 3.7, 3.10, 3.11** | |

<table>
<thead>
<tr>
<th>Module – 3</th>
<th>8 Hours</th>
</tr>
</thead>
</table>
| Fuzzy logic: introduction, human learning ability, undecidability, probability theory, classical set and fuzzy set, fuzzy set operations, fuzzy relations, fuzzy compositions, natural language and fuzzy interpretations, structure of fuzzy inference system, illustrative problems  
**Text Book 1: Chapter 5** | |

<table>
<thead>
<tr>
<th>Module – 4</th>
<th>8 Hours</th>
</tr>
</thead>
</table>
| Introduction to GA, GA, procedures, working of GA, GA applications, applicability, evolutionary programming, working of EP, GA based Machine learning classifier system, illustrative problems  
**Text Book 1: Chapter 7** | |

<table>
<thead>
<tr>
<th>Module – 5</th>
<th>8 Hours</th>
</tr>
</thead>
</table>
| Swarm Intelligent system: Introduction, Background of SI, Ant colony system  
Working of ACO, Particle swarm Intelligence (PSO).  
**Text Book 1: 8.1-8.4, 8.7** | |

**Course outcomes:** The students should be able to:
- Understand soft computing techniques
- Apply the learned techniques to solve realistic problems
- Differentiate soft computing with hard computing techniques

**Question paper pattern:**
The question paper will have ten questions. There will be 2 questions from each module.
Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**
# COMPUTER VISION AND ROBOTICS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

**SEMESTER – VII**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS752</td>
<td></td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

## Module – 1

**CAMERAS:** Pinhole Cameras, **Radiometry – Measuring Light:** Light in Space, Light Surfaces, Important Special Cases, **Sources, Shadows, And Shading:** Qualitative Radiometry, Sources and Their Effects, Local Shading Models, Application: Photometric Stereo, Interreflections: Global Shading Models, **Color:** The Physics of Color, Human Color Perception, Representing Color, A Model for Image Color, Surface Color from Image Color.

**Module – 2**

**Linear Filters:** Linear Filters and Convolution, Shift Invariant Linear Systems, Spatial Frequency and Fourier Transforms, Sampling and Aliasing, Filters as Templates, **Edge Detection:** Noise, Estimating Derivatives, Detecting Edges, **Texture:** Representing Texture, Analysis (and Synthesis) Using Oriented Pyramids, Application: Synthesis by Sampling Local Models, Shape from Texture.

**Module – 3**

**The Geometry of Multiple Views:** Two Views, **Stereopsis:** Reconstruction, Human Stereopsis, Binocular Fusion, Using More Cameras, **Segmentation by Clustering:** What Is Segmentation?, Human Vision: Grouping and Gestalt, Applications: Shot Boundary Detection and Background Subtraction, Image Segmentation by Clustering Pixels, Segmentation by Graph-Theoretic Clustering.

**Module – 4**

**Segmentation by Fitting a Model:** The Hough Transform, Fitting Lines, Fitting Curves, Fitting as a Probabilistic Inference Problem, Robustness, **Segmentation and Fitting Using Probabilistic Methods:** Missing Data Problems, Fitting, and Segmentation, The EM Algorithm in Practice, **Tracking With Linear Dynamic Models:** Tracking as an Abstract Inference Problem, Linear Dynamic Models, Kalman Filtering, Data Association, Applications and Examples.

**Module – 5**

**Geometric Camera Models:** Elements of Analytical Euclidean Geometry, Camera Parameters and the Perspective Projection, Affine Cameras and Affine Projection Equations, **Geometric Camera Calibration:** Least-Squares Parameter Estimation, A Linear Approach to Camera Calibration, Taking Radial Distortion into Account, Analytical Photogrammetry, An Application: Mobile Robot Localization, **Model- Based Vision:** Initial Assumptions, Obtaining Hypotheses by Pose Consistency, Obtaining Hypotheses by pose Clustering, Obtaining Hypotheses Using Invariants, Verification, Application: Registration In Medical Imaging Systems, Curved Surfaces and Alignment.

**Course outcomes:** The students should be able to:
- Implement fundamental image processing techniques required for computer vision
- Perform shape analysis
- Implement boundary tracking techniques
- Apply chain codes and other region descriptors
- Apply Hough Transform for line, circle, and ellipse detections.
- Apply 3D vision techniques.
- Implement motion related techniques.
- Develop applications using computer vision techniques.

**Question paper pattern:**
The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**
# DIGITAL IMAGE PROCESSING

**[As per Choice Based Credit System (CBCS) scheme]**

**(Effective from the academic year 2017 - 2018)**

**SEMESTER – VII**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS753</td>
<td>40</td>
<td>3</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

<table>
<thead>
<tr>
<th>Module – 1</th>
<th><strong>Teaching Hours</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td>Fundamental Steps in Digital Image Processing, Components of an Image Processing System, Sampling and Quantization, Representing Digital Images (Data structure), Some Basic Relationships Between Pixels- Neighbors and Connectivity of pixels in image, Applications of Image Processing: Medical imaging, Robot vision, Character recognition, Remote Sensing.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 2</th>
<th><strong>Teaching Hours</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image Enhancement In The Spatial Domain:</strong></td>
<td>Some Basic Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics of Spatial Filtering, Smoothing Spatial Filters, Sharpening Spatial Filters, Combining Spatial Enhancement Methods.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 3</th>
<th><strong>Teaching Hours</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image Enhancement In Frequency Domain:</strong></td>
<td>Introduction, Fourier Transform, Discrete Fourier Transform (DFT), properties of DFT, Discrete Cosine Transform (DCT), Image filtering in frequency domain.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 4</th>
<th><strong>Teaching Hours</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image Segmentation:</strong></td>
<td>Introduction, Detection of isolated points, line detection, Edge detection, Edge linking, Region based segmentation- Region growing, split and merge technique, local processing, regional processing, Hough transform, Segmentation using Threshold.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module – 5</th>
<th><strong>Teaching Hours</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image Compression:</strong></td>
<td>Introduction, coding Redundancy, Inter-pixel redundancy, image compression model, Lossy and Lossless compression, Huffman Coding, Arithmetic Coding, LZW coding, Transform Coding, Sub-image size selection, blocking, DCT implementation using FFT, Run length coding.</td>
</tr>
</tbody>
</table>

**Course outcomes:** The students should be able to:

- Explain fundamentals of image processing
- Compare transformation algorithms
- Contrast enhancement, segmentation and compression techniques

**Question paper pattern:**

The question paper will have ten questions.

There will be 2 questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

### STORAGE AREA NETWORKS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

#### SEMESTER – VII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS754</td>
<td></td>
<td>3</td>
<td>60</td>
<td>40</td>
<td>3</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

| Module – 1 | **Storage System** | Introduction to evolution of storage architecture, key data center elements, virtualization, and cloud computing. Key data center elements – Host (or compute), connectivity, storage, and application in both classic and virtual environments. RAID implementations, techniques, and levels along with the impact of RAID on application performance. Components of intelligent storage systems and virtual storage provisioning and intelligent storage system implementations. | 8 Hours |
| Module – 2 | **Storage Networking Technologies and Virtualization** | Fibre Channel SAN components, connectivity options, and topologies including access protection mechanism ‘zoning’, FC protocol stack, addressing and operations, SAN-based virtualization and VSAN technology, iSCSI and FCIP protocols for storage access over IP network, Converged protocol FCoE and its components, Network Attached Storage (NAS) - components, protocol and operations, File level storage virtualization, Object based storage and unified storage platform. | 8 Hours |
| Module – 3 | **Backup, Archive, and Replication** | This unit focuses on information availability and business continuity solutions in both virtualized and non-virtualized environments. Business continuity terminologies, planning and solutions, Clustering and multipathing architecture to avoid single points of failure, Backup and recovery - methods, targets and topologies, Data deduplication and backup in virtualized environment, Fixed content and data archive, Local replication in classic and virtual environments, Remote replication in classic and virtual environments, Three-site remote replication and continuous data protection. | 8 Hours |
| Module – 4 | **Cloud Computing Characteristics and benefits** | This unit focuses on the business drivers, definition, essential characteristics, and phases of journey to the Cloud. Business drivers for Cloud computing, Definition of Cloud computing, Characteristics of Cloud computing, Steps involved in transitioning from Classic data center to Cloud computing environment Services and deployment models, Cloud infrastructure components, Cloud migration considerations. | 8 Hours |
| Module – 5 | **Securing and Managing Storage Infrastructure** | This chapter focuses on framework and domains of storage security along with covering security implementation at storage networking. Security threats, and countermeasures in various domains Security solutions for FC-SAN, IP-SAN and NAS environments, Security in virtualized and cloud environments, Monitoring and managing various information infrastructure components in classic and virtual environments, Information lifecycle management (ILM) and storage tiering. | 8 Hours |
**Course outcomes:** The students should be able to:

- Identify key challenges in managing information and analyze different storage networking technologies and virtualization
- Explain components and the implementation of NAS
- Describe CAS architecture and types of archives and forms of virtualization
- Illustrate the storage infrastructure and management activities

**Question paper pattern:**
The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

1. *Information Storage and Management*, Author: EMC Education Services, Publisher: Wiley ISBN: 9781118094839

**Reference Books:**

NIL
MACHINE LEARNING LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)
SEMESTER – VII

Subject Code 17CSL76 IA Marks 40
Number of Lecture Hours/Week 01I + 02P Exam Marks 60
Total Number of Lecture Hours 40 Exam Hours 03

CREDITS – 02

Description (If any):
1. The programs can be implemented in either JAVA or Python.
2. For Problems 1 to 6 and 10, programs are to be developed without using the built-in classes or APIs of Java/Python.
3. Data sets can be taken from standard repositories (https://archive.ics.uci.edu/ml/datasets.html) or constructed by the students.

Lab Experiments:
1. Implement and demonstrate the FIND-Salgorithms for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithms to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
9. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes/API can be used for this problem.
10. Implement the non-parametric Locally Weighted Regressionalgorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

Study Experiment / Project: NIL

Course outcomes: The students should be able to:
1. Understand the implementation procedures for the machine learning algorithms.
2. Design Java/Python programs for various Learning algorithms.
3. Apply appropriate data sets to the Machine Learning algorithms.
4. Identify and apply Machine Learning algorithms to solve real world problems.

<table>
<thead>
<tr>
<th>Conduction of Practical Examination:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• All laboratory experiments are to be included for practical examination.</td>
</tr>
<tr>
<td>• Students are allowed to pick one experiment from the lot.</td>
</tr>
<tr>
<td>• Strictly follow the instructions as printed on the cover page of answer script</td>
</tr>
<tr>
<td>• Marks distribution: Procedure + Conduction + Viva: 15 + 70 + 15 (100)</td>
</tr>
</tbody>
</table>

Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.
WEB TECHNOLOGY LABORATORY WITH MINI PROJECT
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

SEMMESTER – VII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>EAX Marks</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CSL77</td>
<td>40</td>
<td>60</td>
<td>02</td>
</tr>
</tbody>
</table>

Number of Lecture Hours/Week: 01L + 02P

Lab Experiments:

**PART A**

1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.

2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.

3. Write a JavaScript code that displays text “TEXT-GROWING” with increasing font size in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.

4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:
   a. Parameter: A string
   b. Output: The position in the string of the left-most vowel
   c. Parameter: A number
   d. Output: The number with its digits in the reverse order

5. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

6. Write a PHP program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

7. Write a PHP program to display a digital clock which displays the current time of the server.

8. Write the PHP programs to do the following:
   a. Implement simple calculator operations.
   b. Find the transpose of a matrix.
   c. Multiplication of two matrices.
   d. Addition of two matrices.

9. Write a PHP program named states.py that declares a variable states with value "Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following:
   a. Search for a word in variable states that ends in xas. Store this word in element 0 of a list named statesList.
b. Search for a word in states that begins with k and ends in s. Perform a case-insensitive comparison. [Note: Passing re.I as a second parameter to method compile performs a case-insensitive comparison.] Store this word in element 1 of statesList.

c. Search for a word in states that begins with M and ends in s. Store this word in element 2 of the list.

d. Search for a word in states that ends in a. Store this word in element 3 of the list.

10. Write a PHP program to sort the student records which are stored in the database using selection sort.

Study Experiment / Project:

Develop a web application project using the languages and concepts learnt in the theory and exercises listed in part A with a good look and feel effects. You can use any web technologies and frameworks and databases.

Note:

1. In the examination each student picks one question from part A.
2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report (15-20 pages) that must include the following
   a. Introduction
   b. Requirement Analysis
   c. Software Requirement Specification
   d. Analysis and Design
   e. Implementation
   f. Testing

Course outcomes: The students should be able to:

- Design and develop dynamic web pages with good aesthetic sense of designing and latest technical know-how's.
- Understand the concepts of Web Application Terminologies, Internet Tools other web services.
- Recall how to link and publish web sites

Conduction of Practical Examination:

1. All laboratory experiments from part A are to be included for practical examination.
2. Mini project has to be evaluated for 40 Marks.
3. Report should be prepared in a standard format prescribed for project work.
4. Students are allowed to pick one experiment from the lot.
5. Strictly follow the instructions as printed on the cover page of answer script.
6. Marks distribution:
   a) Part A: Procedure + Conduction + Viva: \(09 + 42 + 09 = 60\) Marks
   b) Part B: Demonstration + Report + Viva voce \(20+14+06 = 40\) Marks

Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.
# INTERNET OF THINGS TECHNOLOGY

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

## SEMESTER – VIII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS81</td>
<td>40</td>
<td>60</td>
<td>50</td>
<td>03</td>
</tr>
</tbody>
</table>

## CREDITS – 04

### Module – 1


### Module – 2


### Module – 3


### Module – 4


### Module – 5


### Course Outcomes:

- Interpret the impact and challenges posed by IoT networks leading to new architectural models.
- Compare and contrast the deployment of smart objects and the technologies to connect them to network.
• Appraise the role of IoT protocols for efficient network communication.
• Elaborate the need for Data Analytics and Security in IoT.
• Illustrate different sensor technologies for sensing real world entities and identify the applications of IoT in Industry.

Question paper pattern:

The question paper will have ten questions.
There will be 2 questions from each module.
Each question will have questions covering all the topics under a module.
The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:


Reference Books:

# BIG DATA ANALYTICS

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

## SEMESTER – VIII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
<th>CREDITS – 04</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS82</td>
<td>40</td>
<td>4</td>
<td>60</td>
<td>50</td>
<td>03</td>
<td></td>
</tr>
</tbody>
</table>

## Course outcomes:
The students should be able to:

- Explain the concepts of HDFS and MapReduce framework
- Investigate Hadoop related tools for Big Data Analytics and perform basic Hadoop Administration
- Recognize the role of Business Intelligence, Data warehousing and Visualization in decision making
- Infer the importance of core data mining techniques for data analytics
- Compare and contrast different Text Mining Techniques

## Question paper pattern:
The question paper will have ten questions. There will be 2 questions from each module.
Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

## Text Books:

## Reference Books:
2. Boris Lublinsky, Kevin T.Smith, Alexey Yakubovich, "Professional Hadoop
<table>
<thead>
<tr>
<th>Subject Code</th>
<th>17CS831</th>
<th>IA Marks</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Lecture Hours/Week</td>
<td>3</td>
<td>Exam Marks</td>
<td>60</td>
</tr>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

### Module – 1

**Introduction:** Computational Science and Engineering: Computational Science and Engineering Applications; characteristics and requirements, Review of Computational Complexity, Performance: metrics and measurements, Granularity and Partitioning, Locality: temporal/spatial/stream/kernel, Basic methods for parallel programming, Real-world case studies (drawn from multi-scale, multi-discipline applications)

**Teaching Hours:** 08 Hours

### Module – 2


**Teaching Hours:** 08 Hours

### Module – 3


**Teaching Hours:** 08 Hours

### Module – 4

**Parallel Programming:** Revealing concurrency in applications, Task and Functional Parallelism, Task Scheduling, Synchronization Methods, Parallel Primitives (collective operations), SPMD Programming (threads, OpenMP, MPI), I/O and File Systems, Parallel Matlabs (Parallel Matlab, Star-P, Matlab MPI), Partitioning Global Address Space (PGAS) languages (UPC, Titanium, Global Arrays)

**Teaching Hours:** 08 Hours

### Module – 5

**Achieving Performance:** Measuring performance, Identifying performance bottlenecks, Restructuring applications for deep memory hierarchies, Partitioning applications for heterogeneous resources, using existing libraries, tools, and frameworks

**Teaching Hours:** 08 Hours

### Course outcomes:
The students should be able to:

- Illustrate the key factors affecting performance of CSE applications
- Illustrate mapping of applications to high-performance computing systems
- Apply hardware/software co-design for achieving performance on real-world applications

### Question paper pattern:
The question paper will have ten questions. There will be 2 questions from each module.
Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

**USER INTERFACE DESIGN**  
[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2016 -2017)  

**SEMESTER – VIII**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Number of Lecture Hours/Week</th>
<th>Exam Marks</th>
<th>Total Number of Lecture Hours</th>
<th>Exam Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS832</td>
<td>40</td>
<td>03</td>
<td>60</td>
<td>40</td>
<td>03</td>
</tr>
</tbody>
</table>

**CREDITS – 03**

**Course Objectives:** This course will enable students
- To study the concept of menus, windows, interfaces.
- To study about business functions.
- To study the characteristics and components of windows and the various controls for the windows.
- To study about various problems in window design with text, graphics.
- To study the testing methods.

<table>
<thead>
<tr>
<th>Module –1</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>The User Interface-Introduction, Overview, The importance of user interface – Defining the user interface, The importance of Good design, Characteristics of graphical and web user interfaces, Principles of user interface design.</td>
<td>08 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module –2</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>The User Interface Design process- Obstacles, Usability, Human characteristics in Design, Human Interaction speeds, Business functions-Business definition and requirement analysis, Basic business functions, Design standards.</td>
<td>08 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module –3</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>System menus and navigation schemes- Structures of menus, Functions of menus, Contents of menus, Formatting of menus, Phrasing the menu, Selecting menu choices, Navigating menus, Kinds of graphical menus.</td>
<td>08 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module –4</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows - Characteristics, Components of window, Window presentation styles, Types of window, Window management, Organizing window functions, Window operations, Web systems, Characteristics of device based controls.</td>
<td>08 Hours</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module –5</th>
<th>Teaching Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen based controls- Operable control, Text control, Selection control, Custom control, Presentation control, Windows Tests-prototypes, kinds of tests.</td>
<td>08 Hours</td>
</tr>
</tbody>
</table>

**Course outcomes:** The Students should be able to:
- Design the User Interface, design, menu creation ,windows creation and connection between menus and windows.

**Question paper pattern:**
The question paper will have ten questions. 
There will be 2 questions from each module. 
Each question will have questions covering all the topics under a module. 
The students will have to answer 5 full questions, selecting one full question from each module.

**Text Book:**
**Reference Books:**

| **Module – 2** | Basic Foundations: Standards, Models, and Language: Network Management Standards, Network Management Model, Organization Model, Information Model – Management Information Trees, Managed Object Perspectives, Communication Model; ASN.1- Terminology, Symbols, and Conventions, Objects and Data Types, Object Names, An Example of ASN.1 from ISO 8824; Encoding Structure; Macros, Functional Model. |
**Module – 5**


<table>
<thead>
<tr>
<th>Course outcomes: The students should be able to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Analyze the issues and challenges pertaining to management of emerging network technologies such as wired/wireless networks and high-speed internets.</td>
</tr>
<tr>
<td>• Apply network management standards to manage practical networks</td>
</tr>
<tr>
<td>• Formulate possible approaches for managing OSI network model.</td>
</tr>
<tr>
<td>• Infer SNMP for managing the network</td>
</tr>
<tr>
<td>• Infer RMON for monitoring the behavior of the network</td>
</tr>
<tr>
<td>• Identify the various components of network and formulate the scheme for the managing them</td>
</tr>
</tbody>
</table>

**Question paper pattern:**

The question paper will have ten questions.

There will be 2 questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**


**Reference Books:**

## SYSTEM MODELLING AND SIMULATION

[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 - 2018)

### SEMESTER – VIII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS834</td>
<td></td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Lecture Hours/Week</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Lecture Hours</td>
<td>40</td>
</tr>
<tr>
<td>Exam Hours</td>
<td>03</td>
</tr>
</tbody>
</table>

### CREDITS – 03

### Module – 1

**Introduction:** When simulation is the appropriate tool and when it is not appropriate, Advantages and disadvantages of Simulation; Areas of application, Systems and system environment; Components of a system; Discrete and continuous systems, Model of a system; Types of Models, Discrete-Event System Simulation examples: Simulation of queuing systems. **General Principles, Simulation Software:** Concepts in Discrete-Event Simulation. The Event-Scheduling / Time-Advance Algorithm, Manual simulation Using Event Scheduling

### Module – 2


### Module – 3

**Random-NumberGeneration:** Properties of random numbers; Generation of pseudo-random numbers, Techniques for generating random numbers, Tests for Random Numbers, **Random-Variate Generation:** Inverse transform technique, Acceptance-Rejection technique.

### Module – 4

**Input Modeling:** Data Collection; Identifying the distribution with data, Parameter estimation, Goodness of Fit Tests, Fitting a non-stationary Poisson process, Selecting input models without data, Multivariate and Time-Series input models. **Estimation of Absolute Performance:** Types of simulations with respect to output analysis, Stochastic nature of output data, Measures of performance and their estimation. **Contd..**

### Module – 5

Measures of performance and their estimation, Output analysis for terminating simulations Continued... Output analysis for steady-state simulations. **Verification, Calibration And Validation:** Optimization: Model building, verification and validation, Verification of simulation models, Verification of simulation models, Calibration and validation of models, Optimization via Simulation.

### Course outcomes:
The students should be able to:
- Explain the system concept and apply functional modeling method to model the activities of a static system
- Describe the behavior of a dynamic system and create an analogous model for a dynamic system;
- Illustrate the operation of a dynamic system and make improvement according to the simulation results.

**Question paper pattern:**
The question paper will have ten questions. There will be 2 questions from each module. Each question will have questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module.

**Text Books:**

**Reference Books:**
INTERNSHIP / PROFESSIONAL PRACTISE
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017 -2018)
SEMESTER – VIII

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>IA Marks</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>17CS84</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

Duration 4 weeks Exam Hours 03

CREDITS – 02

Description (If any):

With reference to the above subject, this is to inform that the following are the guidelines to be followed for the Internship Programme and the earlier circular as cited in ref (i) is hereby withdrawn:

1) As per the 15OR 9 the Internship Programme duration is of Eight weeks. However it has been reduced to Four weeks and it should be carried out between (VI and VII Semester) Vacation and/or (VII and VIII Semester) Vacation.

2) The internship can be carried out in any Industry/R and D Organization/Research Institute/ Educational institute of repute.

3) The Institutions may also suggest the students to enrol for the Internshala platform for free internships as there is a MoU with the AICTE for the beneficial of the affiliated Institutions (https://internshala.com/)

4) The Examination of Internship will be carried out in line with the University Project Viva-voce examination.

5) (a) The Department/college shall nominate staff member/s to facilitate, guide and supervise students under internship. (b) The Internal Guide has to visit place of internship at least once during the student’s internship.

6) The students shall report the progress of the internship to the guide in regular intervals and seek his/her advice.

7) After the completion of Internship, students shall submit a report with completion and attendance certificates to the Head of the Department with the approval of both internal and external guides.

8) The Examination of Internship will be carried out in line with the University Project Viva-voce examination.

9) There will be 50 marks for CIE (Seminar: 25, Internship report: 25) and 50 marks for Viva – Voce conducted during SEE. The minimum requirement of CIE marks shall be 50% of the maximum marks.

10) The internal guide shall award the marks for seminar and internship report after evaluation. He/she will also be the internal examiner for Viva – Voce conducted during SEE.

11) The external guide from the industry shall be an examiner for the viva voce on Internship. Viva-Voce on internship shall be conducted at the college and the date of Viva-Voce shall be fixed in consultation with the external Guide. The Examiners shall jointly award the Viva Voce marks.
Course outcomes: The students should be able to:

1. Adapt easily to the industry environment
2. Take part in team work
3. Make use of modern tools
4. Decide upon project planning and financing.
5. Adapt ethical values.
6. Motivate for lifelong learning
PROJECT WORK PHASE II
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2017-2018)

SEMESTER – VIII

Subject Code | 17CSP85 | IA Marks | 100
--- | --- | --- | ---
Number of Lecture Hours/Week | 06 | Exam Marks | 100
Total Number of Lecture Hours | -- | Exam Hours | 03

CREDITS – 06

Description (If any):

- Project: Carried out at the Institution or at an Industry.
- Project work shall preferably be batch wise, the strength of each batch shall not exceed maximum of four students
- Viva-voce examination in project work shall be conducted batch-wise.
- For Project Phase –I and Project seminar and Project Phase –II, the CIE shall be 100 respectively.
- The CIE marks in the case of projects in the final year shall be based on the evaluation at the end of VIII semester by a committee consisting of the Head of the concerned Department and two senior faculty members of the Department, one of whom shall be the project guide.
- Minimum requirement of CIE marks for Project work shall be 50% of the maximum marks.
- Students failing to secure a minimum of 50% of the CIE marks in Project work shall not be eligible for the Project examination conducted by the University and they shall be considered as failed in that/those Course/s. However, they can appear for University examinations conducted in other Courses of the same semester and backlog Courses if any. Students after satisfying the prescribed minimum CIE marks in the Course/s when offered during subsequent semester shall appear for SEE.
- Improvement of CIE marks shall not be allowed in Project where the student has already secured the minimum required marks
- For a pass in a Project/Viva-voce examination, a student shall secure a minimum of 40% of the maximum marks prescribed for the University Examination. The Minimum Passing Grade in a Course is ‘E’.
- The student who desires to reject the results of a semester shall reject performance in all the Courses of the semester, irrespective of whether the student has passed or failed in any Course. However, the rejection of performance of VIII semester project shall not be permitted

Course outcomes: The students should be able to:

1. Identify a issue and derive problem related to society, environment, economics, energy and technology
2. Formulate and Analyze the problem and determine the scope of the solution chosen
3. Determine, dissect, and estimate the parameters, required in the solution.
4. Evaluate the solution by considering the standard data / Objective function and by using appropriate performance metrics.
5. Compile the report and take part in present / publishing the finding in a reputed conference / publications
6. Attempt to obtain ownership of the solution / product developed.
SEMESTER – VIII

Subject Code 17CSS86 IA Marks 100
Number of Lecture Hours/Week 04 Exam Marks --
Total Number of Lecture Hours -- Exam Hours --

CREDITS – 01

Description:

- Seminar: Deliverable at the Institution under the supervision of a Faculty.
- Seminar is one of the head of passing. i) Each candidate shall deliver seminar as per the Scheme of Teaching and Examination on the topics chosen from the relevant fields for about 30 minutes. ii) The Head of the Department shall make arrangements for conducting seminars through concerned faculty members of the Department. The committee constituted for the purpose by the Head of the Department shall award the CIE marks for the seminar. The committee shall consist of three faculty from the Department and the senior most acting as the Chairman/Chairperson. [To be read along with 17 OB 8.6]
- For Technical seminar, the CIE marks shall be 100.
- The CIE marks in the case of projects and seminars in the final year shall be based on the evaluation at the end of VIII semester by a committee consisting of the Head of the concerned Department and two senior faculty members of the Department, one of whom shall be the project / seminar guide.
- For seminar, the minimum requirement of CIE marks shall be 40% of the maximum marks.
- If any student fails to secure a minimum of 40% of the maximum CIE marks in seminar/ fails to deliver the seminar, he/she shall be considered as failed in that Course and shall not be eligible for the award of degree. However, the student shall become eligible for the award of degree after satisfying the requirements prescribed for seminar during the subsequent semester/s.
- Improvement of CIE marks shall not be allowed in Seminar where the student has already secured the minimum required marks.
- Seminar topics must be from recent advancements in the domain.
- Each candidate must submit three copies of the report to the department. One for the candidate, one for the guide and one for the department.

Course outcomes: The students should be able to:

- Survey the changes in the technologies relevant to the topic selected
- Discuss the technology and interpret the impact on the society, environment and domain.
- Compile report of the study and present to the audience, following the ethics.