PREFACE

Dear Students,

From the academic year 2014-15 there is a slight change in the syllabus structure and question paper pattern. This change is due to the philosophy of Outcome Based Education and requirement as per the National Board of Accreditation (NBA), Government of India, New Delhi.

Sixteen countries including New Zealand, Australia, Singapore, Russia and India are the signatories of the Washington Accord, which has come out with the new process of accreditation. This would enable every institution, including NIE to attain high standards of technical education in the respective countries and to create level playing ground. The outcome based education is one of the important components of NBA.

NIE is making sincere efforts in meeting the global standards through new formats of NBA and timely World Bank-MHRD initiative TEQIP (Technical Education Quality Improvement Program). Efforts are being made to revise the syllabi regularly to meet the challenges of the current technical education.

Dr. B. K. Sridhara
Dean (Academic Affairs)

July 2014
BLUEPRINT OF SYLLABUS STRUCTURE AND QUESTION PAPER PATTERN

(to be effective from the odd semester of the academic year 2014-15 for all semester students)

Blue Print of Syllabus Structure

1. Complete syllabus is prescribed in SIX units as Unit 1, Unit 2, etc.

2. In each unit there is one topic under the heading “Self Learning Exercises” (SLE). These are the topics to be learnt by the student on their own under the guidance of the course instructors. Course instructors will inform the students about the depth to which SLE components are to be studied. Thus there will be six topics in the complete syllabus which will carry questions with a weightage of 10% in SEE only. No questions will be asked on SLE components in CIE.

Blue Print of Question Paper

1. Question paper will have seven full questions.

2. One full question each of 15 marks (Question No 1, 2, 3, 4, 5 and 6) will be set from each unit of the syllabus. Out of these six questions, two questions will have internal choice from the same unit. The unit in which choice is to be given is left to the discretion of the course instructor.

3. Question No 7 will be set for 10 marks only on those topics prescribed as “Self Learning Exercises”.

Dr. B. K. Sridhara
Dean (Academic Affairs)

July 2014
### Scheme of Teaching and Examination
#### Department of Information Science and Engineering
#### Fifth Semester – B.E

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Subject Code</th>
<th>Course Title</th>
<th>Teaching Dept</th>
<th>Contact Hours / Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IS0410</td>
<td>Microprocessor</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>IS0411</td>
<td>System Software</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>IS0412</td>
<td>Data Mining</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>IS0413</td>
<td>Data Communication</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>IS0414</td>
<td>Operating System</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>IS0201</td>
<td>Introduction to Engineering &amp; Design</td>
<td>IS &amp; E</td>
<td>L:2 T:0 P:0 Total:2</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>IS0106</td>
<td>Microprocessor Lab</td>
<td>IS &amp; E</td>
<td>L:0 T:0 P:3 Total:3</td>
<td>1.5</td>
</tr>
<tr>
<td>8</td>
<td>IS0107</td>
<td>System Software Lab</td>
<td>IS &amp; E</td>
<td>L:0 T:0 P:3 Total:3</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>IS04xx</td>
<td>Elective – I</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>L:28 T:0 P:8 Total:32</td>
<td>29</td>
</tr>
</tbody>
</table>

Pattern of course evaluation for both CIE and SEE will be mentioned in the abridged lesson plan and the Course Instructor (CI) will discuss the same with the students during the first/second session of the semester.

### Scheme of Teaching and Examination
#### Department of Information Science and Engineering
#### Sixth Semester – B.E

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Subject Code</th>
<th>Course Title</th>
<th>Teaching Dept</th>
<th>Contact Hours / Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IS0415</td>
<td>Software Engineering</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>IS0416</td>
<td>Web Programming</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>IS0417</td>
<td>Unix Systems Programming</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>IS0418</td>
<td>Computer Networks</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>IS0108</td>
<td>Web Programming Lab</td>
<td>IS &amp; E</td>
<td>L:0 T:0 P:2 Total:2</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>IS0109</td>
<td>Unix Systems Programming Lab</td>
<td>IS &amp; E</td>
<td>L:0 T:0 P:2 Total:2</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>IS0110</td>
<td>Computer Networks Lab</td>
<td>IS &amp; E</td>
<td>L:0 T:0 P:2 Total:2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>IS04xx</td>
<td>Elective – II</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>IS04xx</td>
<td>Elective – III</td>
<td>IS &amp; E</td>
<td>L:4 T:0 P:0 Total:4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td></td>
<td>L:24 T:0 P:6 Total:30</td>
<td>27</td>
</tr>
</tbody>
</table>

Pattern of course evaluation for both CIE and SEE will be mentioned in the abridged lesson plan and the Course Instructor (CI) will discuss the same with the students during the first/second session of the semester.
V SEMESTER
MICROPROCESSOR (4:0:0)

Sub Code : IS0410        CIE : 50% Marks
Hrs/Week : 04               SEE : 50% Marks
SEE Hrs : 03 Hrs          Max. Marks : 100

Course Outcomes:
On Successful completion of the course, the students will be able to:
1. Describe the internal architecture of 8086.
2. Explain the program structures in assembly language.
3. Use the string instruction, Procedure, Macro to perform operations on a sequence of data words in memory.
4. Describe the system with bus protocol during Read/Write machine cycle.
5. Explain 8086 Interrupts and Interrupt Applications.
6. Describe digital Interfacing.

UNIT 1: 9 Hrs
An overview of computer and microprocessor, 8086 Family Assembler Language Programming: Architecture of computer, Advances in computer architecture, RISC and CISC, 8086 Internal Architecture, Introduction to programming the 8086. Constructing the machine codes for 8086 Instructions.
Self Learning Exercise: Assembly language program development tools.

UNIT 2: 9 Hrs
8086 Instruction Descriptions and Assembly Directives: Instruction Description, Assembler Directives, Assume, DB, DD, DQ, DT, DW, END, ENDP, ENDS, EQU, EVEN, EXTRN, INCLUDE, LABEL, LENGTH, NAME, OFFSET.
Self Learning Exercise: ORG, PROC, PTR, Segment, Short, Type.

UNIT 3: 8 Hrs
Implementing Standard Program Structures in 8086 Assembly Language: Simple Sequence Programs, Jumps, Flags, and Conditional Jumps, If-Then, If-Then-Else, and Multiple If-Then-Else Programs, While-Do Programs, Repeat-Ultil Programs, Instruction Timing and Delay Loops, Strings, Procedures, and Macros: The 8086 String Instructions.
Self Learning Exercise: Writing and Using Procedures, Writing and Using Assembler Macros

UNIT 4: 8 Hrs
8086 System connection Timing: A basic 8086 microprocessor system, 8086 pin diagram/description, main factor of 8086, Minimum mode and Maximum mode configuration. Self Learning Exercise: Activities during Read/Write cycle.

UNIT 5: 9 Hrs
8086 Interrupts and Interrupt Applications: 8086 Interrupts and Interrupt Responses, Hardware Interrupt Applications, 8259A Priority Interrupt Controller.
Self Learning Exercise: Software Interrupt Applications.

UNIT 6: 9 Hrs
Digital Interfacing: Programmable Parallel Ports and Handshake Input / Output: Methods of parallel data transfer, Implementing handshake data transfer, 8255 internal block diagram and system connection, operational mode and initialization, constructing and sending 8255 control word, Interfacing Microprocessor to keyboard
Self Learning Exercise: D / A convertor operation, Interfacing and application.

TEXT BOOK:

**REFERENCE BOOKS:**

SYSTEM SOFTWARE (4:0:0)

Sub Code : IS0411  
CIE : 50% Marks  
Hrs/Week : 04  
SEE : 50% Marks  
SEE Hrs : 03 Hrs  
Max. Marks : 100

Course Outcomes:
On Successful completion of the course, the students will be able to:

1. Define the features of system software.
2. Design assemblers with the basic features for SIC architecture.
3. Distinguish between machine independent and machine specific details of assemblers.
4. Discuss the working of loaders and linkers.
5. Explain the various phases of compiler and features of macroprocessors.
6. Solve the problems like token reorganization and validating grammar using Lex and YACC tools.

UNIT 1: 9 Hrs
Introduction:
Characteristics of system software, System software as software supporting the operation and use of the computer itself. Design and implementation of system software.

Machine Architecture:
The Simplified Instructional Computer (SIC) – SIC Machine Architecture.


UNIT 2: 8 Hrs
Assemblers – 1:
Basic Assembler Function – A Simple SIC Assembler, Assembler Algorithm and Data Structures.
Machine Dependent Assembler Features – Instruction Formats & Addressing Modes,

Self Learning Exercise: Program Relocation.

UNIT 3: 9 Hrs
Assemblers – 2:
Machine Independent Assembler Features – Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking.
Assembler Design Operations – One-Pass Assembler.
Self Learning Exercise: Multi-Pass Assembler.

UNIT 4: 9 Hrs
Machine-Dependent Loader Features – Relocation, Program Linking
Self Learning Exercise: Algorithm and Data Structures for a Linking Loader.

UNIT 5: 8 Hrs
Macro Processor: Basic Macro Processor Functions – Macro Definitions and Expansion.
Compilers: Basic compiler functions, Grammars, Lexical analysis, modeling scanners as finite state automata, Syntactic analysis.

UNIT 6: 9 Hrs
The Definition Section, The Rules Section, Symbol Values and Actions.

**Self Learning Exercise:** The LEXER, Compiling and Running a Simple Parser

**TEXT BOOK:**


**REFERENCE BOOK:**

DATA MINING (4:0:0)

Sub Code : IS0412 CIE : 50% Marks
Hrs/Week : 04 SEE : 50% Marks
SEE Hrs : 03 Hrs Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Describe different types to data preprocessing.
2. Discuss general approaches to solve classification problem.
3. Apply association rule to discover relationship in large data sets.
4. Distinguish different types of clusters.
5. Identify approaches to anomaly detection.
6. Discuss applications and trends in data mining.

UNIT 1: 8 Hrs
Introduction:
What is Data Mining? Motivating Challenges, The origins of data mining, Data Mining Tasks. Types of Data, Data Quality. Data Preprocessing
Self Learning Exercise: Measures of Similarity and Dissimilarity.

UNIT 2: 9 Hrs
Classification:
Preliminaries, General approach to solving a classification problem, Decision tree induction, Rule-based classifier
Self Learning Exercise: Nearest-neighbor classifier.

UNIT 3: 9 Hrs
Association Analysis:
Problem Definition, Frequent Itemset generation, Rule Generation, Compact representation of frequent itemsets, Alternative methods
for generating frequent itemsets. FP-Growth algorithm, Evaluation of association patterns, Effect of skewed support distribution

**Self Learning Exercise:** Sequential patterns.

**UNIT 4:**

9 Hrs

**Cluster Analysis:**
Overview of cluster analysis, K-means, Agglomerative hierarchical clustering, DBSCAN.

**Self Learning Exercise:** Overview of Cluster Evaluation, Cluster Analysis: Example-comparing K-means and DBSCAN.

**UNIT 5:**

9 Hrs

**Anomaly Detection:**
Preliminaries; Statistical approaches, Proximity-based outlier detection, Density-based outlier detection.

**Self Learning Exercise:** Cluster-based techniques.

**UNIT 6:**

8 Hrs

**Applications:**
Data mining applications, Data mining system products and research prototypes, Additional themes on Data mining.

**Self Learning Exercise:** Social impact of Data mining, Trends in Data mining.

**TEXT BOOK:**

1. *Introduction to Data Mining*, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison-Wesley 2006.

2. *Data Mining – Concepts and Techniques*, Jiawei Han, Micheline Kamber, 2nd Edition, Morgan Kaufmann, 2005

**REFERENCE BOOK:**

1. *Insight into Data Mining - Theory and Practice*, K.P.Soman, Shyam Diwakar, V.Ajay.
DATA COMMUNICATION (4:0:0)

Sub Code : IS0413        CIE : 50% Marks
Hrs/Week : 04           SEE : 50% Marks
SEE Hrs : 03 Hrs        Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Distinguish between OSI and TCP/IP model.
2. Describe digital signal and different analog-to-digital conversion techniques for transmission.
3. Describe digital-to-analog conversion techniques for transmission with multiplexing.
4. Discuss error detection and correction techniques, Hamming distance.
5. Discuss the data link layer protocols and framing.
6. Discuss different Multiple Access & IEEE Standards.

UNIT 1: 8 Hrs
Introduction:
Self Learning Exercise: Addressing – Physical, Logical and Port addressing.

UNIT 2: 9 Hrs
Data, Signals and Digital Transmission:
Analog and digital, Digital signals, Transmission impairment, Data rate limits, Performance, Digital-to-Digital conversion, Analog-to-Digital conversion – PCM, DM.
Self Learning Exercise: Transmission modes.
UNIT 3: 9 Hrs
Analog Transmission and Multiplexing:
Digital-to-Analog conversion – ASK, FSK, PSK, QAM, Analog-to-Analog conversion – AM, FM, PM, Multiplexing – FDM, TDM, WDM
Self Learning Exercise: Spread spectrum – FHSS, DSSS.

UNIT 4: 8 Hrs
Error Detection and Correction:
Introduction to error detection correction, Block coding – Error detection, correction, hamming distance, Min Hamming distance, Linear block codes
Self Learning Exercise: Cyclic codes, Checksum.

UNIT 5: 9 Hrs
Data Link Control:
Framing; Flow and Error control, Protocols, Noiseless channels – Simplest, Stop and Wait protocol, Noisy channels – Stop and Wait ARQ, Go Back N ARQ, Selective Repeat ARQ, Piggybacking, HDLC, Point-to-point Protocol – Framing.
Self Learning Exercise: Transition phases.

UNIT 6: 9 Hrs
Multiple Access, Ethernet, Wireless Lan’s:
Self Learning Exercise: IEE 802.11, Bluetooth.

TEXT BOOK:
REFERENCE BOOK:


OPERATING SYSTEM (4:0:0)

Sub Code : IS0414  
CIE : 50% Marks

Hrs/Week : 04  
SEE : 50% Marks

SEE Hrs : 03 Hrs  
Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Distinguish between process and threads
2. Discuss inter-process communication
3. Differentiate between paging and segmentation
4. Explain file system implementation
5. Distinguish between deadlock avoidance and deadlock prevention.
6. Differentiate between monolithic and layered systems

UNIT 1:  
Introduction, Processes and Threads:  


Threads: The Thread Model, Thread Usage, Implementing Threads in User Space, Implementing Threads in the Kernel, Hybrid Implementations, Scheduler Activations.

Scheduling: Introduction to Scheduling, Scheduling in Batch Systems, Scheduling in Interactive Systems, Scheduling in Real-Time Systems

Self Learning Exercise: Policy versus Mechanism, Thread Scheduling.
UNIT 2: Interprocess Communication: 9 Hrs
Race Conditions, Critical Regions, Mutual Exclusion with Busy Waiting, Sleep and Wakeup, Semaphores, Mutexes, Monitors, Message Passing.
Memory Management
Basic Memory Management: Monoprogramming without Swapping or Paging, Multiprogramming with Fixed Partitions, Modeling Multiprogramming, Analysis of Multiprogramming System Performance, Relocation and Protection.
Swapping: Memory Management with Bitmaps, Memory Management with Linked Lists.
Virtual Memory: Paging, Page Tables

UNIT 3: Replacement Algorithms and Design Issues: 8 Hrs
Design Issues For Paging Systems: Local versus Global Allocation Policies, Load Control, Page Size, Separate Instruction and Data Spaces, Shared Pages, Cleaning Policy, Virtual Memory Interface.
Implementation Issues: Operating System Involvement with Paging, Page Fault Handling, Instruction Backup, Locking Pages in Memory, Backing Store, Separation of Policy and Mechanism.
Segmentation: Implementation of Pure Segmentation  
Self Learning Exercise: Segmentation with Paging.

UNIT 4:  
9 Hrs  
Input / Output:  
I/O Software Layers: Interrupt Handlers, Device Drivers, Device-Independent I/O Software.  
Clocks: Clock Hardware, Clock Software.  
File Systems:  
File System Implementation: File System Layout, Implementing Files, Implementing Directories, Shared Files, Disk Space Management, File System Reliability  

UNIT 5:  
9 Hrs  
Deadlocks:  
Introduction to Deadlocks: Conditions for Deadlock, Deadlock Modeling.  
The Ostrich Algorithm, Deadlock Detection And Recovery: Deadlock Detection with One Resource of Each Type, Deadlock Detection with Multiple Resource of Each Type, Recovery from Deadlock.  
Self Learning Exercise: Attacking the No Preemption Condition, Attacking the Circular Wait Condition.
UNIT 6: 8 Hrs

**System Calls:** System Calls for Process Management, System Calls for File Management, System Calls for Directory Management, Miscellaneous System Calls.

**Operating System Structure:** Monolithic Systems, Layered Systems, Virtual Machines, Exokernels, Client-Server Model.

**Self Learning Exercise:** Example File Systems: The UNIX File System.

**TEXT BOOK:**

**REFERENCE BOOK:**
INTRODUCTION TO ENGINEERING AND DESIGN (2:0:0)

Sub Code : IS0201  
CIE : 50% Marks

Hrs/Week : 02  
SEE : 50% Marks

SEE Hrs : 02 Hrs  
Max. Marks : 50

Course Outcomes

On Successful completion of the course, the students will be able to:

1. Distinguish between personal and professional ethics.
2. Analyze the importance of units in engineering analysis.
3. List steps involved in problem solving process.
4. Discuss the various steps needed in engineering design problem solving.
5. Analyze the performance evaluation.
6. Discuss the different guidelines in preparing the report.

UNIT 1: 4 Hrs

What Engineers do


Key Elements of engineering analysis

Engineering analysis, The SI Unit system, Force

Self Learning Exercise: Weight and Mass, Significant figures

UNIT 2: 4 Hrs

Engineering Economics

Why Is Economics Important? The Cost of Money, When Is an Investment Worth It?

Introduction to Engineering Design

The Nature of Engineering Design, Design Problems versus Homework Problems, Benefits of a Hands-On Design Project, Qualities of a Good Designer, How to Manage a Design Project,
Two Grounds Rules for Design, The Need for a Systematic Approach

**Self Learning Exercise:** Steps in the Engineering Design Process, Hands-On Design Exercise: The Tower

**UNIT 3:**
5 Hrs

**Design Step 1**
Defining the Problem
Problem Definition, List of Specifications, Design Milestone: Clarification of the Task

**Design Step 2**
**Generation of Alternative Concepts**
Brainstorming, Concept Sketching, Hands-On Design Exercise: The Tube, Research-Based Strategies for Promoting Creativity
Functional Decomposition for Complex Systems

**Self Learning Exercise:** Design Milestone: Generation of Alternatives.

**UNIT 4:**
5 Hrs

**Design Step 3**
**Evaluation of Alternatives Concepts of A Concept**

**Design Step 4**
Detailed Design
Analysis, Experiments, Models, Detailed Drawings

**Self Learning Exercise:** Design Milestone: Detailed Design

**UNIT 5:**
4 Hrs

**Design Step 5**
Design Defense

**Design Step 6**
Manufacturing and Testing
Manufacturing and Testing Strategies, Materials, Joining Methods, Useful Hand Tools, Design Milestone: Design for Manufacture Assessment I

Self Learning Exercise: Design Milestone: Design for Manufacture Assessment II.

UNIT 6: 4 Hrs
Design Step 7
Performance Evaluation:

Design Step 8
Design Report
Organization of the Report, Writing Guidelines

Self Learning Exercise: Design Milestone: Design Report

TEXT BOOK:
MICROPROCESSOR LAB (0:0:3)

Sub Code : IS0106                Max. Marks : 50
Hrs/Week : 03

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Write a program using program structures such as if-else, multiple if-then-else and so on.
2. Write a program using 8086 instructions such as add, mul, div and so on.
3. Implement the instruction like DAA, XLAT using procedure/subroutines.
4. Write interrupt service procedure.
5. Write a program for keyboard detect and determine row and column information.
6. Write a program for BCD up/down counter and Ring counter using logic controller.

PART – A

1. Write separate ALPs to add, to subtract and to find an average of two numbers.
2. Write an ALP to check given number is positive or negative.
3. Write an ALP to display message on the screen using Macros.
4. Write an ALP to find the largest of N numbers.
5. Write an ALP to find whether the given string is palindrome or not.
6. Write an ALP to perform binary search and display the output on the monitor.
PART – B

7. Read status of eight input bits from the Logic Controller Interface and display FF if it is even parity bits otherwise display 00. Also display number of 1,s in the input data.
8. Perform the following functions using the Logic Controller Interface.
   a. BCD up-down counter
   b. Ring counter
   c. Jonson counter
9. Display message FIRE and HELP alternately with flickering effects on a seven segment display interface for a suitable period of time.
10. Scan 3X8 Keypad for key closure and to store the code of the key pressed in a memory location or display it on the screen. Also display row and column of the key pressed.
11. Write a program to drive an Elevator interface to service multiple requests.
12. Using the D/A converter generate
   a. Rectangular waveform with varying duty cycle for On and Off. (High or Low)
   b. Triangular waveform with varying peak value and base value.

TEXT BOOK:

REFERENCE BOOKS:
   (Section 4-7, Section 8-1).
SYSTEM SOFTWARE LAB (0:0:3)

Sub Code : IS0107  Max. Marks : 50
Hrs/Week : 03

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Solve the problems like token reorganization and validating grammar using Lex and YACC tools
2. Design assemblers with the basic features for SIC architecture.

PART – A

1. Write a c program for file handling.
2. Write a c program to print command line arguments.
3. Write a lexer in C to recognize an identifier with the usual definition. Underscores are allowed (not to begin and not to end).

LEX:

1. Program to count the number of vowels and consonants in a given string.
2. Program to count the number of characters, words, spaces, end of lines in a given input file.
3. Program to count the no of ‘scanf’ and ‘printf’ statements in a C program. Replace them with ‘readf’ and ‘writef’ statements respectively.

YACC:

1. Program to test the validity of a simple expression involving operators +, -, *, and /
2. Program to recognize nested IF control statements and display the levels of nesting.
3. Program to recognize strings ‘aaab’, ‘abbb’, ‘ab’ and ‘a’ using grammar (a^n b^n, n>=0)
PART – B

MINI PROJECT:

1. Develop pass-1 of assembler to generate the intermediate file. You may assume fixed format or free format for the assembly language statement. First develop and test the following functions that are required for the development of a 2-pass assembler.
   - read_next_inputline(...) (reads an entire line that may have a label, op-code(mnemonic), operand and comment.
   - get_label (from the input line)
   - get_op_code(from the input line)
   - get_operand(from the input line)
   - Search_symbol_table_for_label(...)
   - insert_label_locctr_into_symbol_table(...)
   - Search_op_code_table_for_opcode
   - write_line_to_intermediate_file(..)

2. Develop pass-2 phase of the assembler to generate the object file by making use of the intermediate file generated by pass-1 and functions developed.

TEXT BOOK:

VI SEMESTER
SOFTWARE ENGINEERING (4:0:0)

Sub Code : IS0415  CIE : 50% Marks
Hrs/Week : 04  SEE : 50% Marks
SEE Hrs : 03 Hrs  Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Classify the software process and their development models.
2. Explain the software practices efficiently.
3. Discuss the steps involved in software requirements gathering.
4. Discuss the software design methods and different testing strategies.
5. Explain the estimation and scheduling of the project.
6. Explain the project management and metrics involved in software development.

UNIT 1: 8 Hrs
Self Learning Exercise: Process technology, Product and process.

UNIT 2: 9 Hrs
Process models, Software engineering practice and System Engineering: Prescriptive models, waterfall model, Incremental process models, Evolutionary process models, specialized process

**Self Learning Exercise:** Business process engineering, product engineering and system modeling.

**UNIT 3:** 9 Hrs

**Requirement and Design engineering:** A Bridge to design and construction, requirement engineering tasks, initiating the requirements engineering process, Eliciting requirements, negotiating requirements, validating requirements. Design with the context of software engineering, design process and design quality, design concepts.

**Self Learning Exercise:** Design model, pattern based software

**UNIT 4:** 9 Hrs

**Architectural design, Modeling Component level design, Testing strategies and Testing tactics:** Software design, data design, Architectural styles and patterns, Architectural design. What is a component?, Designing class based components. A strategic approach to software engineering, strategic issues, test strategies for conventional software. Software testing fundamentals, Black box and white box testing, White box testing- Basis path testing, control structures testing. Black box testing

**Self Learning Exercise:** Testing for specialized environments, architectures and applications.

**UNIT 5:** 9 Hrs

**Estimation and Project Scheduling:** Observation on estimation, Project planning process, Software scope and feasibility, Resources, Software project estimation, decomposition techniques, and Empirical estimation models. Basic concepts, project scheduling, defining a task set for the software project.

**Self Learning Exercise:** Defining a task network, Schedule.
UNIT 6:  8 Hrs
Project Management and Metrics for process and projects:
The Management spectrum, The people, the product, the process, the project, metrics for product and process. Metrics in process and project domains.
Self Learning Exercise: Software measurements, metrics for software quality.

TEXT BOOK:

REFERENCE BOOK:
WEB PROGRAMMING (4:0:0)

Sub Code : IS0416  CIE : 50% Marks
Hrs/Week : 04  SEE : 50% Marks
SEE Hrs : 03 Hrs  Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Create web pages using HTML, XHTML.
2. Use Cascading Style Sheets and operators, variables, arrays, control structures, functions, objects in JavaScript.
5. Create Common Gateway Interface (CGI) program using PERL scripting language.
6. Design web applications using PHP. (Server Side Programming)

UNIT 1: 8 Hrs
Fundamentals of Web: Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security.
Self Learning Exercise: Tables.

UNIT 2: 8 Hrs
Introduction to XHTML(Cont.): Forms, The Audio Element, The Video Element, Organization Elements, The Time Element, Syntactic differences between HTML and XHTML.
Cascading Style Sheets: Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms,
Font, List properties, Color, Alignment of text, The box model, Background images.

**Self Learning Exercise:** The `<span>` and `<div>` tags, Conflict resolution.

**UNIT 3:**

**Basics of JavaScript:** Overview of JavaScript, Object orientation and JavaScript, Syntactic characteristics, Primitives, operations, and expressions, Screen output and keyboard input, Control statements, Object creation and modification, Arrays, Functions, Constructors, Pattern matching using regular expressions, Errors in scripts, Examples.

**JavaScript and XHTML Documents:** The JavaScript execution environment, The Document Object Model, Element access in JavaScript, Events and event handling, handling events from the Body elements, Button elements, Text box and Password elements, The DOM 2 event model, The canvas Element

**Self Learning Exercise:** The navigator object, DOM tree traversal and modification.

**UNIT 4:**

**Introduction to XML:** Introduction, Syntax, Document structure, Document type definitions, Namespaces, XML schemas, displaying raw XML documents, Displaying XML documents with CSS, XSLT style sheets

**Self Learning Exercise:** XML processors, Web services.

**UNIT 5:**

**Perl:** Introduction, Perl Documentation, Perl Syntax Rules, A Quick Introduction To Object-Oriented Programming, Database Independent Interface.

**The Common Gateway Interface:** Introduction, Apache Configuration, A First CGI Program, What Can Go Wrong?, CGI.pm Introduced, CGI.pm HTML Shortcuts, Information Received by the CGI Program, Form Widget Methods
Self Learning Exercise: CGI Security Considerations, A Note About die(), Project-CGI/MySQL/

UNIT 6: 9 Hrs
Introduction to PHP: Origins and uses of PHP, Overview of PHP, General syntactic characteristics, Primitives, operations and expressions, Output, Control statements, Arrays, Functions, Pattern matching, Form handling, Files.
Self Learning Exercise: Cookies, Session tracking, Database access with PHP and MySQL.

TEXT BOOK:
2. Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP, James Lee & Brent Ware. (UNIT – 5)

REFERENCE BOOK:
UNIX SYSTEMS PROGRAMMING (4:0:0)

Sub Code : IS0417  CIE : 50% Marks
Hrs/Week : 04  SEE : 50% Marks
SEE Hrs : 03 Hrs  Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:
1. Describe the basic features of unix OS.
2. Illustrate system calls.
3. Discuss the processes in unix environment.
4. Identify the importance of process control and its relationship.
5. Apply signals and daemon concepts in unix environment.
6. Demonstrate inter-process communication in unix.

UNIT 1: 8 Hrs
Self Learning Exercise: Feature Test Macros, Primitive System Data Types, Conflicts Between Standards

UNIT 2: 9 Hrs

**Self Learning Exercise:** Device Special Files, Summary of File Access Permission Bits.

**UNIT 3:**


**Self Learning Exercise:** getrlimit, setrlimit Functions, UNIX Kernel Support for Processes.

**UNIT 4:**


**Self Learning Exercise:** Shell Execution of Programs, Orphaned Process Groups.

**UNIT 5:**

**Self Learning Exercise:** Coding Rules, Single-instance daemons, Daemon conventions.

**UNIT 6:**  
**Interprocess Communication, Network IPC and Sockets:**  
Introduction, Pipes, popen, pclose Functions, Coprocesses, FIFOs, Message Queue, Semaphores. Introduction, Socket Descriptors, Addressing.  
**Self Learning Exercise:** Connection establishment, Data transfer, Socket options.

**TEXT BOOK:**

**REFERENCE BOOK:**
2. *Advanced Unix Programming*, Marc J Rochkind, 2\textsuperscript{nd} Edition.  
COMPUTER NETWORKS (4:0:0)

Sub Code : IS0418
Hrs/Week : 04
SEE Hrs : 03 Hrs
CIE : 50% Marks
SEE : 50% Marks
Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Explain the concepts of IP address, Classes, IP Datagram’s.
2. Discuss the mapping of logical to physical addresses and vice versa and also use the different protocols for allocating IP addresses.
3. Discuss and Differentiate between forwarding and routing algorithms.
4. Discuss the different protocols for process to process communication with flow and error control.
5. Discuss the various congestion control algorithms and also QoS mechanisms.
6. Discuss the importance of various applications related services and protocols.

UNIT 1: 8 Hrs
Network Layer –I (Logical Addressing, Internet Protocol)
Self Learning Exercise: Fragmentation, Checksum, Options.

UNIT 2: 9 Hrs
Network Layer –II (Address Mapping, Error Reporting and Multicasting)
Address Mapping: Mapping Logical to Physical Address: ARP, Mapping Physical to Logical Address: RARp, BOOTP, and DHCP. ICMP: Types of Messages, Message Format, Error Reporting messages, Query messages. IGMP: Group Management, IGMP Messages.

**Self Learning Exercise:** Message Format, IGMP Operation, Encapsulation.

**UNIT 3:**

**Network Layer –III (Delivery, Forwarding and Routing)**


**Self Learning Exercise:** Distance Vector Routing, Link State Routing.

**UNIT 4:**

**Transport Layer – I (Process-to-Process Delivery: UDP, TCP, and SCTP)**


**Self Learning Exercise:** A TCP Connection, Flow Control, SCTP features.

**UNIT 5:**

**Transport Layer - II (Congestion Control and Quality of Service)**

Classes. Techniques To Improve QoS: Scheduling, Traffic Shaping.

Self Learning Exercise: Resource Reservation, Admission Control.

UNIT 6: 9 Hrs  
Application Layer  
Domain Name System: Name Space: Flat Name Space, Hierarchical Name Space. Domain Name Space: Label, Domain Name, Domain. Distribution Of Name Space: Hierarchy of Name Servers, Zone, Root Server, Primary and Secondary Servers. DNS in the Internet: Generic Domains, Country Domains, Inverse Domain.  

TEXT BOOK:  

REFERENCE BOOK:  
WEB PROGRAMMING LAB (0:0:2)

Sub Code : IS0108  Max. Marks : 50
Hrs/Week : 02

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Create web pages using HTML, XHTML and Cascading Styles sheets.
3. Build web applications using PHP, Perl and CGI.(server side programming)

List of Programs:
1. Develop and demonstrate a XHTML document that illustrates the use external style sheet, ordered list, table, borders, padding, color, and the <span> tag.
2. Develop and demonstrate a XHTML file that includes Javascript for the following problems:
   a) Input: A number n obtained using prompt
      Output: The first n Fibonacci numbers
   b) Input: A number n obtained using prompt
      Output: A table of numbers from 1 to n and their squares using alert
3. Develop and demonstrate a XHTML file that includes Javascript script that uses functions for the following problems:
   a) Parameter: A string
      Output: The position in the string of the left-most vowel
   b) Parameter: A number
      Output: The number with its digits in the reverse order
4. a) Develop and demonstrate, using Javascript, a XHTML document that collects the USN (the valid format is: A digit
from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.
b) Modify the above program to get the current semester also (restricted to be a number from 1 to 8)

5. a) Design an XML document to store information about a student in an engineering college. The information must include USN, Name, Name of the College, Branch, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.

6. a) Write a Perl program to display various Server Information like Server Name, Server Software, Server protocol, CGI Revision etc.
b) Write a Perl Program to accept three no.s from command prompt and find the largest number using functions.

7. a) Write a Perl program to accept UNIX command from a HTML form and to display the output of the command executed.
b) Write a Perl program to accept a number from a HTML form and to check whether it is a prime no. or not and display the output on browser using HTML Shortcut methods.

8. a) Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.
b) Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

9. a) Write a Perl program to display a greeting based on the access time of the web server. Also to verify whether the webmaster is currently logged in or not.
b) Write a Perl program to display a digital clock which displays the current time of the server.

10. Write a Perl program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.

11. a) Write a PHP program to store current date-time in a COOKIE and display the ‘Last visited on’ date-time on the web page upon reopening of the same page.  
b) Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.

12. Using PHP and MySQL, develop a program to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

Open Ended Experiments:
Students should develop a website about their college / library or any other problem which includes all the features studied in the course.

TEXT BOOK:

REFERENCE BOOK:
UNIX SYSTEMS PROGRAMMING LAB (0:0:2)

Sub Code : IS0109  Max. Marks : 50
Hrs/Week : 02

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Create simple programs using shell script.
2. Illustrate system calls using files and directories.
3. Demonstrate inter-process communication using C programs.

List of Programs:

1. Write a shell script to check whether a given number is positive or negative.
2. Shell scripts: Create a file and write all UNIX commands that you want to execute and save that file with extension .sh.
3. File management system calls: Write a program to implement
   a. Create a file
   b. Copy one file to another
   c. Linking a file
   d. Delete a file.
4. Directory management system calls: Write a program to change current working directory and display the inode details for each file in the new directory.
5. Parent process – Child process Relationship. Write a program that creates a zombie, and then call system to execute the ps(1) command to verify that the process is a zombie.
6. Implementing IPC using pipes: Write a program that creates a child process. Parent process writes data to pipe and child process reads the data from pipe and prints it on the screen.
7. Write a C program that illustrates the following:
   a. Creating a message queue.
   b. Writing to a message queue.
   c. Reading from a message queue.
8. Write a C program that illustrates inter process communication using shared memory system calls.
9. Implementation of semaphore: Write a C program that demonstrates how two processes can share a variable using semaphore.
10. Write a C program that implements a producer-consumer system with two processes. (using semaphores)
11. Write a C program that illustrates file locking using semaphores.
12. Implementation of shell: Write a 'c' program to implement a shell.
13. Write a client program and a server program to return the number of processes currently running on a specified host computer.
14. Write two library routines, one to enable asynchronous I/O on a socket and one to disable asynchronous I/O on a socket.

TEXT BOOK:

REFERENCE BOOK:
COMPUTER NETWORKS LAB (0:0:2)

Sub Code : IS0110       Max. Marks : 50
Hrs/Week : 02

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Demonstrate the computer network concepts using a simulator.
2. Implement the concepts of communication among the computers in a network using C/C++ programming.

PART – A

Simulation Exercises:
The following experiments shall be conducted using NCTUNS / OPNET / NS2 or any other suitable simulator.

1. Simulate a three node point-to-point network with duplex links between them. Set the queue size and vary the bandwidth and find the number of packets dropped.

2. Simulate a four node point-to-point network with the links connected as follows:
n0 – n2, n1 – n2 and n2 – n3. Apply TCP agent between n0-n3 and UDP between n1-n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets sent by TCP / UDP.

3. Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

4. Simulate an Ethernet LAN using n nodes (6-10), change error rate and data rate and compare throughput.

5. Simulate an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.
6. Simulate simple ESS and with transmitting nodes in wireless LAN by simulation and determine the performance with respect to transmission of packets.

**PART – B**

**Implement the following in C/C++.**

7. Write a program for error detecting code using CRC-CCITT (16-bit).

8. 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.

9. Implement the above program using message queues or FIFOs as IPC channels.

10. Write a client / server program wherein the client sends the host name and the server returns the ip address. Implement it using a connectionless service.

11. Write a program for Hamming Code generation for error detection and Correction.

12. Write a program to simulate ARP/RARP.

**Open Ended Exercises:**

- To know and learn about various network related commands [ping, tracert, netstat, at, net, route, arp, nbtstat, traceroute, ifconfig] and few definitions cum settings.
- Create different topologies.

**TEXT BOOK:**


**REFERENCE BOOK:**


47
ELECTIVES

MANAGEMENT AND INFORMATION SYSTEMS (4:0:0)

Sub Code : IS0423
Hrs/Week : 04
SEE Hrs : 03 Hrs

CIE : 50% Marks
SEE : 50% Marks
Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Discuss the potential of Information Systems.
2. Examine how to Communicate and manage data through the use of personal productivity software tools.
3. Analyze the issues of Supply chain and N/W.
4. Identify the concepts of Artificial Intelligence and expert systems.
5. Apply the ethical responsibilities and cyber laws and identify the challenges faced in social issues.
6. Analyze the cultural, political and geo-economic challenges in global management of IT.

UNIT 1: 8 Hrs

Foundations of Information Systems in Business:

UNIT 2: 9 Hrs
- **Business Systems and Functional Business Systems:**
- **Self Learning Exercise:** Accounting Systems, Financial Management Systems.

UNIT 3: 9 Hrs
- **Enterprise Business Systems:**
  - **Customer Relationship Management:**
    - Introduction, What is CRM? Real World Case 1, The Three Phases of CRM, Benefits and Challenges of CRM, Trends in CRM.
  - **Enterprise Resource Planning:**
    - Introduction, What is ERP? Benefits and Challenges of ERP, Trends in ERP.
  - **Supply Chain Management:**
    - Introduction, What is SCM? The Role of SCM.
- **Self Learning Exercise:** Benefits and Challenges of SCM, Trends in SCM.

UNIT 4: 9 Hrs
- **Decision Support Systems, Decision Support in Business:**
UNIT 5: 9 Hrs
Security and Ethical Challenges, Security, Ethical and Social challenges of IT:

Security Management of IT:

Self Learning Exercise: Other Security Measures, System Controls and Audits.

UNIT 6: 8 Hrs
Enterprise and Global Management of IT:
Managing IT and Managing Global IT: Business and IT, Managing IT, Business/IT Planning, Managing the IT Function, Organizing IT, Outsourcing and Offshoring IT and IS, Failures in IT management. The International Dimension, Global IT Management, Cultural, Political and Geo - Economic Challenges, Global Business/ IT strategies, Global Business / IT Applications, Global IT Platforms

Self Learning Exercise: Global Data Access Issues, Global Systems Development

TEXT BOOK:

REFERENCE BOOK:
COMPILER DESIGN (4:0:0)

Sub Code : IS0426  CIE : 50% Marks
Hrs/Week : 04  SEE : 50% Marks
SEE Hrs : 03 Hrs  Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Analyze the basic functioning of compiler and Design and implement finite state machines appropriate for TINY Language lexical scanner.
2. Define the grammar and semantics of a language.
3. Design a top down parser for the definition of an appropriate context free grammar.
4. Explain the semantic definitions for an appropriate language.
5. Discuss the various Run Time Environment used during program execution.
6. Illustrate various Intermediate code generation algorithms for a compiler and Perform code generation at the tuple level.

UNIT 1: 9 Hrs
Introduction and Scanners:

UNIT 2: 8 Hrs
Context-Free Grammars and Parsing:

UNIT 3: 9 Hrs
Top-Down Parsing:
Top-Down Parsing by Recursive-Descent, LL(1) Parsing, First and Follow Sets.
Self Learning Exercise: A Recursive-Descent Parser for the TINY Language, Error Recovery.

UNIT 4: 9 Hrs
Semantic Analysis:
Attributes and Attribute Grammars, Algorithms for Attribute Computation, The Symbol Table, Data Types and Type Checking
Self Learning Exercise: A Semantic Analyzer for the TINY Language.

UNIT 5: 8 Hrs
Runtime Environments:
Memory Organization During Program Execution, Fully Static Runtime Environments, Stack-Based Runtime Environments, Dynamic Memory, Parameter Passing Mechanisms.
Self Learning Exercise: Runtime Environment for the TINY Language.

UNIT 6: 9 Hrs
Code Generation:
TEXT BOOK:


REFERENCE BOOK:

DISTRIBUTED OPERATING SYSTEMS (4:0:0)

Sub Code : IS0427
Hrs/Week : 04
SEE Hrs : 03 Hrs

CIE : 50% Marks
SEE : 50% Marks
Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Describe categories of distributed and parallel computer architectures independent of circuit packaging.
2. Compare and contrast various distributed operating system designs based upon resource sharing, reliability, and performance design goals.
3. Give an account of the theoretical models used to design distributed systems and to manipulate those models to reason about such systems.
4. Describe system architecture and operating design parameters that significantly impact algorithm performance.
5. The student will understand and estimate the impact of different design choices, system features on distributed systems.
6. Use modern operating system techniques and tools for the design of multi-processing and multi-threaded applications.

UNIT 1: Introduction to Distributed Systems: 9 Hrs
What is a distributed system, Goals, Hardware and Software Concepts.
Self Learning Exercise: Design issues.

UNIT 2: Communication in Distributed Systems: 8 Hrs
Asynchronous transfer mode networks, Client – Server Model, Remote Procedure Call.
Self Learning Exercise: Group Communication.

UNIT 3: 9 Hrs
Synchronization in Distributed Systems:
Clock synchronization, Mutual exclusion, Election algorithms, Atomic transactions.
Self Learning Exercise: Deadlocks in Distributed Systems.

UNIT 4: 9 Hrs
Processes and Processors in Distributed Systems:
Threads, System Models, Processor Allocation, Scheduling in distributed systems, Fault tolerance
Self Learning Exercise: Real time distributed systems.

UNIT 5: 8 Hrs
Distributed File systems:
Distributed file system design, Distributed file system implementation.
Self Learning Exercise: Case study

UNIT 6: 9 Hrs
Distributed Shared Memory:
What is shared memory, Consistency models, Page based distributed shared memory, Shared variable distributed shared memory
Self Learning Exercise: Object based distributed shared memory.

TEXT BOOK:

REFERENCE BOOK:
MULTIMEDIA COMPUTING (4:0:0)

Sub Code : IS0428  
CIE : 50% Marks

Hrs/Week : 04  
SEE : 50% Marks

SEE Hrs : 03 Hrs  
Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Discuss types of multimedia networks and multimedia applications
2. Compare the different types of document formats
3. Design the different types of text compression techniques
4. Apply different types of compression techniques on audio and images
5. Analyze the different types of video compression techniques
6. Compare the different standards for the multimedia communications

UNIT 1:  
Multimedia communications

8 Hrs

Introduction; Multimedia information representation; Multimedia networks: Telephone networks, Data networks, Broadcast television networks, Integrated services digital networks, Broadband multiservice networks; Multimedia applications: Interpersonal communications, Interactive applications over the Internet, Entertainment applications; Application and networking terminology: Media types, Communication modes, Network types.

Self Learning Exercise: Multipoint conferencing, Network QoS, Application QoS.

UNIT 2:  
Multimedia information representation

9 Hrs

Introduction; Digitization principles: Analog signals, Encoder design, Decoder design; Text: Unformatted text, Formatted text,
Hypertext; Images: Graphics, Digitized documents, Digitized pictures; Audio: PCM speech, CD-quality audio, synthesized audio.

**Self Learning Exercise:** Video: Broadcast television, Digital video, PC video, Video content.

**UNIT 3:**

**9 Hrs**

**Compression Techniques (Text, Image, Audio and Video):**
Introduction; Compression principles: Source encoders and destination decoders, Lossless and lossy compression, Entropy encoding, Source encoding.

**Text compression:** static Huffman coding, Dynamic Huffman coding, Arithmetic coding.

**Self Learning Exercise:** Lempel-Ziv coding, Lempel-Ziv-Welsh coding.

**UNIT 4:**

**9 Hrs**

**Image compression:** Graphics interchange format, Tagged image file format, Digitized documents, Digitized pictures, JPEG.

**Audio compression:**

**Self Learning Exercise:** MPEG audio coders, Dolby audio coders.

**UNIT 5:**

**9 Hrs**

**Video compression:** Video compression principles, H.261, H.263, MPEG, MPEG-1.

**Self Learning Exercise:** MPEG-2, MPEG-4.

**UNIT 6:**

**8 Hrs**

**Standards for multimedia communications**
Introduction; Reference models: TCP/IP reference model, protocol basics; Standards relating to interpersonal communications: Circuit-mode networks, packet-switched networks, Electronic mail; Standards relating to interactive applications over the Internet,
Information browsing, Electronic commerce, Intermediate systems, java and JavaScript.

**Self Learning Exercise:** Standards for entertainment applications: Movie/video-on-demand, Interactive television.

**TEXT BOOKS:**


**REFERENCE BOOKS:**

SYSTEM SIMULATION AND MODELING (4:0:0)

Sub Code : IS0429 CIE : 50% Marks
Hrs/Week : 04 SEE : 50% Marks
SEE Hrs : 03 Hrs Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Determine when simulation is appropriate tool and also its advantages and Disadvantages.
2. Understand general principles of simulation.
3. Describe the Characteristics of Queuing Systems.
4. Appreciate random number and random variate generation.
5. Gain knowledge on data collection and selecting input models and appreciate different types of simulation with respect to output analysis.
6. Understand verification and validation of simulation models.

UNIT 1: 9 Hrs
Introduction to Simulation:
When Simulation is the appropriate tool, When Simulation 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.
Self Learning Exercise: Steps in a Simulation study.

UNIT 2: 9 Hrs
Simulation Examples, General Principles and Simulation Software: Simulation of Queuing Systems, Simulation of Inventory Systems, Other Examples of Simulation.

Self Learning Exercise: Simulation in GPSS.

UNIT 3: 9 Hrs
Queuing Model:
Characteristics of Queuing Systems, Queuing Notation

UNIT 4: 9 Hrs
Random-Number Generation, Random-Variate Generation:
Self Learning Exercise: Discrete Distribution, Acceptance-Rejection Technique.

UNIT 5: 9 Hrs
Input Modeling:
Data Collection, Identifying the Distribution with Data, Parameter Estimation, Goodness-of-Fit Tests, Fitting a Nonstationary Poisson Process.
Self Learning Exercise: Selecting Input Models without Data.

UNIT 6: 9 Hrs
Verification and Validation of Simulation Models, Output Analysis for a Single Model:
Model Building, Verification and Validation, Verification of Simulation Models, Calibration and Validation of Models.Types of Simulation with Respect to Output analysis.
TEXT BOOK:

REFERENCE BOOK:
COMPUTER GRAPHICS AND VISUALIZATION (4:0:0)

Sub Code : IS0430 CIE : 50% Marks
Hrs/Week : 04 SEE : 50% Marks
SEE Hrs : 03 Hrs Max. Marks : 100

Course Outcomes
On Successful completion of the course, the students will be able to:

1. Explain the basic components in computer graphic system.
2. Identify and Illustrate the OpenGL APIs with examples.
3. Identify and illustration of Interactive Programs and Animating Interactive Programs. Design of Interactive Programs.
4. Explain and identify basics of Geometric Objects and Transformations
5. Explain, identify and illustrate Geometric Objects and Transformations in Homogeneous Coordinates using OpenGL.
6. Analyze and compare Classical and computer viewing.

UNIT 1: 8 Hrs
Introduction to Images and Graphics Programming:

Self Learning Exercise: The OpenGL API, Primitives and attributes.

UNIT 2: 9 Hrs
The Opengl, Input and Interaction:Color, Viewing, Control functions, The Gasket program, Polygons and recursion, The three-dimensional gasket, Plotting implicit functions. Interaction,
Input devices, Clients and servers, Display lists, Display lists and modeling, Programming event-driven input, Menu, Picking.

**Self Learning Exercise:** Design of interactive programs, Logic operations.

**UNIT 3:** 9 Hrs

**Geometric Objects and Transformations – 1:**
Scalars, points, and vectors, Three-dimensional primitives, Coordinate systems and frames, Modeling a colored cube, Affine transformations.

**Self Learning Exercise:** Rotation, translation and scaling.

**UNIT 4:** 8 Hrs

**Geometric Objects and Transformations – 2:**
Transformations in homogeneous coordinates, Concatenation of OpenGL transformation matrices.

**Self Learning Exercise:** Quaternion.

**UNIT 5:** 9 Hrs

**Viewing:**

**Self Learning Exercise:** Perspective-projection matrices, Projections and shadows.

**UNIT 6:** 9 Hrs

**Lighting and Shading:**

**Self Learning Exercise:** Shading of the sphere model, Global illumination.
TEXT BOOK:

REFERENCE BOOK: