### DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

### SCHEME OF TEACHING AND EXAMINATION

#### B.E. (ISE) V SEMESTER
**(ACADEMIC YEAR 2015-16)**

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Subject Code</th>
<th>Subject</th>
<th>Credits</th>
<th>Hours/Week</th>
<th>Examination Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lecture</td>
<td>Tutorial</td>
</tr>
<tr>
<td>1</td>
<td>UIS501C</td>
<td>System Software</td>
<td>04</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>UIS522C</td>
<td>Software Engineering</td>
<td>04</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>UIS503C</td>
<td>Database Management Systems</td>
<td>04</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>UIS505C</td>
<td>Java Programming</td>
<td>04</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>UIS00XX</td>
<td>Elective – I</td>
<td>03</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>UIS00XX</td>
<td>Elective – II</td>
<td>03</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>UIS515L</td>
<td>System Software Laboratory</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>UIS506L</td>
<td>Microprocessor Laboratory</td>
<td>1.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>UIS510L</td>
<td>Java Programming Lab</td>
<td>1.5</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

|       |              |                                | 26      | 21         | 3         | 6         | 400 | 400 | 800   |

* Advanced Mathematics-II is a mandatory subject only for students having Diploma and admitted to 3\textsuperscript{rd} Semester through lateral entry Scheme. Passing the subject is compulsory; however marks will not be considered for awarding grade/class. APP/NP grade will be awarded for passing/not passing the subject.

** The total lecture hours for students having Diploma and admitted to 3\textsuperscript{rd} Semester through lateral entry scheme is 26 hours.
### DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

#### SCHEME OF TEACHING AND EXAMINATION

**B.E. (ISE) VI SEMESTER**  
*(ACADEMIC YEAR 2015 - 16)*

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Subject Code</th>
<th>Subject</th>
<th>Credits</th>
<th>Hours/Week</th>
<th>Examination Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Lecture Tutorial</td>
</tr>
<tr>
<td>1</td>
<td>UIS606C</td>
<td>Web Technologies</td>
<td>04</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>UIS603C</td>
<td>Computer Networks.</td>
<td>04</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>UIS604C</td>
<td>Unix System Programming.</td>
<td>04</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>UIS605H/</td>
<td>Professional Communication &amp; Technical Writing</td>
<td>02</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>UIS00XX</td>
<td>Elective-III</td>
<td>03</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>UIS00XX</td>
<td>Elective – IV</td>
<td>03</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>UIS608L</td>
<td>Unix System Programming Laboratory</td>
<td>1.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>UIS619L</td>
<td>Database Application Laboratory</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>UIS611L</td>
<td>Web Programming Lab</td>
<td>1.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td><strong>25</strong></td>
<td><strong>21</strong></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>450</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>450</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>900</strong></td>
</tr>
</tbody>
</table>
V SEMESTER
UIS501C: SYSTEMS SOFTWARE
4-credits (4–0– 0)

UNIT - 1


UNIT - 2


UNIT - 3

UNIT - 4
EDITORS AND DEBUGGING SYSTEMS: Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems - Debugging Functions and Capabilities, Relationship with Other Parts of The System, User-Interface Criteria.

13 Hours

TEXT BOOK:

REFERENCE BOOK:
UNIT- I

INTRODUCTION: Evolution- from an art form to an engineering discipline, software development projects, exploratory style of software development, emergence of software engineering, notable changes in software development practices, computer systems engineering.

SOFTWARE LIFE CYCLE MODELS: A few basic concepts, waterfall model and its extensions, rapid application development, agile development models, spiral model, a comparison of different life cycle models

REQUIREMENT ANALYSIS AND SPECIFICATION: Requirement gathering and analysis, software requirements specification (SRS), formal system specification, Axiomatic specification, algebraic specifications,

UNIT- II

SOFTWARE DESIGN: Overview of the design process, how to characterize a good software design, cohesion and coupling, layered arrangement of Modules, approaches to software design

FUNCTION-ORIENTED SOFTWARE DESIGN: Overview of SA/SD methodology, structured analysis, developing the DFD model of the system, structured design, detailed design, design review

OBJECT MODELLING USING UML: Basic Object-orientation concepts, Unified Modelling Language, UML diagrams, Use case model,, Class diagrams, Interaction diagrams, Activity diagram, State chart Diagram

USER INTERFACE DESIGN: Characteristics of a good user interface, basic concepts, types of user interfaces

UNIT- III

CODING AND TESTING: Introduction to program testing, Coding, code review, software documentation, testing, unit testing, black – box testing, White – box testing, debugging, program analysis tools, integration testing, testing object-oriented programs, systems testing

SOFTWARE RELIABILITY AND QUALITY MANAGEMENT: Software reliability, statistical testing, software quality, software quality management system, ISO 9000, SEI capability maturity model

COMPUTER AIDED SOFTWARE ENGINEERING: CASE and its scope, Case Environment, CASE support in software life cycle, other characteristics of CASE tools

UNIT- IV

SOFTWARE PROJECT MANAGEMENT: software project management complexities, responsibilities of a software project manager, project planning, metrics for project size estimation, project estimation techniques, COCOMO – a heuristic estimation technique, Halstead’s software science an analytic technique, Staffing level estimation, scheduling, organization and team structures, staffing, risk management, software configuration management

EMERGING TRENDS: client- server software, client server architectures, CORBA, COM/DCOM, Service - oriented architecture (SOA), software as a service (SaaS),

Text book:
Reference books:

1. Ian Somerville, Software Engineering, 7th edition, Pearson Education
UIS503C: DATABASE MANAGEMENT SYSTEMS
4-Credits (4-0-0)

UNIT – 1
INTRODUCTION: Introduction; An example; Characteristics of database approach; Advantages of using DBMS approach; when not to use a DBMS. Data models, schemas and instances; Three-schema architecture and data independence; Database languages and interfaces; The database system environment; Centralized and client-server architectures; Classification of Database Management systems.  

ENTITY-RELATIONSHIP MODEL: Using High-Level Conceptual Data Models for Database Design; An example database application; Entity types, Entity sets, Attributes and Keys; Relationship types, Relationship sets, Roles and Structural constraints; Weak entity types; Refining the ER Design; ER Diagrams, Naming conventions and design issues; Relationship types of degree higher than two.  

UNIT – 2
RELATIONAL MODEL AND RELATIONAL DATABASE CONSTRAINTS: Relational model concepts; Relational model constraints and Relational database schemas; Update operations, Transaction and dealing with constraint violations.

RELATIONAL ALGEBRA: Unary relational operations: SELECT and PROJECT; Relational algebra operations from set theory; Binary relational operations: JOIN and DIVISION; Additional relational operations; Examples of queries in relational algebra; Relational database design using ER-to-Relational mapping.

SQL: data definition and data types; Specifying basic constraints in SQL; Schema change statements in SQL; Basic queries in SQL; More complex SQL queries. Insert, Delete and Update statements in SQL; Specifying constraints as Assertion and Trigger; Views (Virtual Tables) in SQL;

UNIT – 3
DATABASE DESIGN: Informal design guidelines for relation schemas; Functional dependencies; Normal forms based on primary keys; General definitions of second and third normal forms; Boyce-Codd Normal Form

Properties of relational decompositions; Algorithms for relational database Schema design; Multivalued dependencies and Fourth Normal Form; Join Dependencies and Fifth Normal Form; Inclusion Dependencies; Other Dependencies and Normal forms.

UNIT – 4
TRANSACTION MANAGEMENT : Introduction to transaction processing; Transaction & system concepts; Desirable properties of transactions; Characterizing schedules based on recoverability; Characterizing schedules based on serializability; Transaction support in SQL;

CONCURRENCY CONTROL: Two-phase locking techniques for concurrency control;

CRASH RECOVERY: Recovery concepts; Recovery techniques based on deferred update; recovery techniques based on immediate update; shadow paging; The ARIES recovery algorithm;

6 hours

6 hours

3 hours

6 hours

5 hours

6 hours

7 hours

8 hours

5 hours
Text book:

Reference books:
UIS505C: JAVA Programming  
CREDITS (4-0-0)  
Unit-I

Introduction to Java:  
History of Java; Java changed the internet: Applets; Byte code: JVM; Java features, JDK, Evolution of Java;  
Object Oriented Programming: Abstraction, Encapsulation, Inheritance and Polymorphism; Simple java programs; Lexical issues;  
Data types: Integers, Floating point types. Characters, Booleans; Literals: Integer literals, Floating point literals, Character literals, Boolean literals, String literals; Variables; Type Conversion and Casting: Java automatic conversion, Casting incompatible casts, Automatic type promotion in expression;  
Arrays: One dimensional arrays, Multi dimensional arrays, Strings;  
Operators: Arithmetic operators, Bitwise operators, Relational operators, Boolean Logical operators, Assignment operators, The? Operator, Operator precedence;  
Control statements: Selection statements, Iteration statements, Jump statements.  

13 Hours

Unit-II

Classes, Inheritance and Exceptions:  
Classes: Defining classes, Declaring Objects, Assigning object reference variables, Defining methods; Constructors: Default constructors, Parameterize constructors, ‘this’ key word; Overloading: Overloading methods, Overloading constructors; Access control; Understanding static; Introducing final, Nested and inner classes; exploring the string class, using command line arguments  
Inheritance: Super classes, Sub classes, Member access, Using super, Multi level inheritance, Method over riding; Dynamic method dispatch, abstract classes using final with inheritance  
Exceptions: Fundamentals, Exception types, Uncaught exceptions, Using try and catch, Multiple catch clauses, Nested try statements, throw, throws.  

13 Hours

Unit-III

Packages, Interfaces and Applets:  
Packages: Defining packages, Finding packages and CLASSPATH, Access protection, importing packages;  
Interfaces: Defining interfaces, Implementing interfaces, Nested interfaces, Applying interfaces, Variables in interfaces;  

13 Hours

Unit-IV

Multithreaded programming and Managing Files:  
Multithreaded programming: Java thread model, Main thread, Creating a thread, Creating multiple threads, Using isArive() and join(), Thread priorities, Synchronization, Interthread communication, Suspending, Resuming and stopping threads, Using multi threading;  
Managing input/output files in Java: Java I/O classes and interfaces, Files, Stream classes, Byte Streams, Random access file, Character streams, Serialization, Stream benefits.  

13 Hours

Text Books:  
Reference Books:  
1. Java for programmers, Paul J. Deitel and Harvey M. Deitel, Pearson Education.  
PART A

Execution of the following programs using C:

1) Write a C program to extract label, opcode and operands given any SIC/XE instruction belonging to any one of the formats discussed below;
   F1: label opcode
   F2: label opcode r1 r2
   F3: label opcode displacement

2) Write a C program translate the given arithmetic statement into SIC/XE assembly code.
   Example: \( C = x+y \) must be translated to;
   LDA x
   ADD y
   STA C

3) Write a C program to generate symbol table of assembler.

4) Write a C program to machine code an instruction belonging to any one of the formats discussed below;
   F1: label opcode
   F2: label opcode r1 r2
   F3: label opcode displacement

5) Write a c-program to implement lexical analysis phase of a compiler.

PART B

Execution of the following programs using LEX:

1) Program to count the number special symbols of C program.
2) Program to count number of
   a) Positive and negative integers
   b) Positive negative fractions
3) Program to count the numbers of comment lines in a given C program. Also eliminate them and copy that program into separate file.
4) Program to count the number of ‘scanf’ and ‘printf’ statements in a C program. Replace them with ‘readf’ and ‘writef’ statements respectively.
5) Program to recognize a valid arithmetic expression and identify the identifiers and operators present. Print them separately.
Execution of the following programs using YACC:

1) Program to recognize nested IF control statements and display the number of levels of nesting.
2) Program to recognize a valid arithmetic expression that uses +, -, *, and /.
3) Program to recognize a valid variable, which starts with a letter, followed by any number of letters or digits.
4) Program to evaluate an arithmetic expression involving operators +, -, *, and / for integers and real data
5) Program to recognize the given the words such as ‘0’ is bounded by equal No. of ‘1’ towards left and right.
   Ex: 101, 11011, 1110111 valid
   110, 01, 1101, 1011 are invalid.

Instructions:
In the examination, a combination of one question from Part A for 20 marks and one LEX and one YACC problem from Part B for 20 Marks has to be given and Viva must be conducted for 10 Marks.

General Remarks:

1) Lab schedule: 3hrs/week for each student (1 hr tutorial, 2 hrs- program execution).
2) Student should complete all the lab assignments.
3) Evaluation CIE 50 marks:
   Lab assignments : 30 marks
   Lab CIE : 10 marks
Part A: Assembly Language Assignments.
1. Write an assembly language program to simulate an arithmetic calculator to support addition, subtraction, multiplication and division on given two 8-bit numbers.
2. Write an assembly language program to add two 3x3 matrices, where matrices are stored in array (row wise).
3. Write an assembly language program to read an alphanumeric character and display its equivalent ASCII code at specified location (x and y coordinate values are given).
4. Write an assembly language program using macros as follows:
   i) To read a character from the keyboard in the module (1)
   ii) To display a character in module (2)
   iii) Use the above two modules to read a string of characters from the keyboard terminated by the carriage return and print the string on the display in the next line.
5. Write an assembly language program to search a key element in a list of 'n' 16-bit numbers using Binary search algorithm.
6. Write an assembly language program to reverse a given string and check whether it is palindrome or not.
7. Write an assembly language program to generate the first N Fibonacci numbers using procedures.
8. Write an assembly language program to create a file (Input File) and to delete an existing file.

Part B: Interfacing Assignments.
1. Write an assembly language program to read the 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.
2. Write an assembly language program to perform the following functions using logic controller interface.
   i) BCD up-down counter
   ii) Ring counter
3. Write an assembly language program to display FIRE and HELP alternatively 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.
4. Write an assembly language program to drive a stepper motor interface to rotate the motor by N steps left direction and N steps right direction. Introduce suitable delay between successive steps.
5. Write an assembly language program to scan a 8*3 keypad for key closure and to store code of key pressed in memory location or display on screen. Also display row and column numbers of key pressed.
6. Write an assembly language program to generate SINE WAVE using DAC (Digital to Analog Converter) interface (the output of the DAC is to be displayed on the CRO).
7. Write an assembly language program to drive an elevator interface in the following way:
   i) Initially the elevator should be in the ground floor, with all requests in OFF state.
   ii) When a request is made from a floor, the elevator should move to that floor, wait there for a couple of seconds, and then come down to ground floor and stop. If any requests arrive while elevator is moving up or coming down they should be ignored.

General Remarks:
1) Lab schedule: 3hrs/week for each student (1 hr tutorial, 2 hrs-program execution).
2) Student should complete all the lab assignments.
3) Evaluation CIE 50 marks:
   Lab assignments: 30 marks
   Lab CIE: 20 marks
1. Write a Java program to read the two matrices A (m x n) & B (x x y) and find the product matrix C (m x y).

2. Write a Java program to perform arithmetic operations on two complex numbers and print the resultant complex number.

3. The annual examination results of N students are tabulated as follows:

<table>
<thead>
<tr>
<th>Roll. No</th>
<th>Subject</th>
<th>Subject 2</th>
<th>Subject 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Write a Java program to read the data and determine
   a) Total marks obtained by each student
   b) The highest marks in each subject & the roll number of the student who scored it
   c) The student who obtained the highest total marks.

4. Write a Java program to sort an array of strings

5. Create a class ACCOUNT that stores customer name, account number, and type of account. From this derive the classes savings account and current account. The savings account provides compound interest and withdrawal facility but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance & if the balance falls below this level, a service charge is imposed. Write a Java program to achieve the following tasks:
   a) Accept deposit from a customer & update the balance
   b) Display the balance
   c) Compute & deposit interest
   d) Permit withdrawal & update the balance
   e) Check for the minimum balance, impose penalty, if necessary and update the balance

6. An educational institute wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are as shown in below figure. Write a Java program to specify all the classes & define methods to create the database and retrieve individual information & when required.

7. Design a package to contain class EMPLOY which holds the information of employees such as Employ number, Name, Department, and Designation, another package to contain Basic salary, Allowance & Deductions. Write a Java program to input class obtained in these two
packages & bring employ number, name, department, designation, basic salary, allowance & deduction along with net salary.

8. Write a Java program to implement the stack operations by defining an interface for stack & a class that implement the stack.

9. Develop an applet that receives three numeric values as input from the user & then displays the largest of the three on the screen. Write a HTML page & test the applet.

10. Write a Java program to print “mythead1” and “mythead2” in parallel by creating thread object.

11. Write a Java program to read string from the keyboard terminated by the symbol $ and write this stream to the filename, r.dat and display the largest character among the string, find total number of characters and total number of words in the stream from the file r.dat.

12. Write a Java program with multiple threads to perform

\[ P = \sin(x) + \cos(y) + \tan(z) \]

**General Remarks:**

1) Lab schedule: 3hrs/week for each student (1 hr tutorial, 2 hrs- program execution).
2) Student should complete all the lab assignments.
3) Evaluation CIE 50 marks:
   - Lab assignments : 30 marks
   - Lab CIE : 20 marks
VI Semester
Web Technologies (UIS 606C) – 4 Credits

UNIT – I

FUNDAMENTALS OF WEB, XHTML - Internet, HTTP request and HTTP response phase, MIME, The Web Programmers Toolbox.

XHTML: Basic syntax; Standard XHTML document structure; Basic text markup. XHTML : Images; Hypertext Links; Lists; Tables; Forms; Frames; Syntactic differences between HTML and XHTML. CSS: Introduction; Levels of style sheets; Style specification formats; Selector forms; Property value forms; CSS: Font properties; List properties; Color; Alignment of text; Background images; The <span> and <div> tags;

Basics of JavaScript: General syntactic characteristics; Primitives, Screen output and keyboard input; Control statements; Object creation and modification, Arrays; Functions; Pattern matching using regular expressions.


13 HRS

UNIT – II


4 HRS


9 HRS

UNIT – III

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, Cookies, Session Tracking, Database access with PHP and MySQL.

13 HRS

UNIT – IV

Ruby and Rails: Origins and uses of Ruby, Scalar types and their operations, Simple input and output, Control statements, Arrays, Hashes, Methods, Classes, Code blocks and iterators, Pattern matching. Overview of Rails, Document requests, processing forms, Rails applications with Databases, Layouts.

13 HRS

TEXT BOOK:

REFERENCE BOOKS:
UIS603C: COMPUTER NETWORKS
4 CREDITS (4-0-0)

Unit-I:
Introduction::Data Communications: Definition and characteristics, Components, Data representations, Data flow, Networks: Definition, Distributed Processing, Network Criteria, Physical structures, Categories of Networks [LAN, WAN, MAN], Protocols: Key elements. The OSI Model: layered architecture, peer to peer processes, and encapsulation, Layers in the OSI model [Brief description of all seven layers], TCP / IP Protocol Suite [Brief description]: physical, data link, transport and application layer, Addressing: physical, logical and port addresses.


12 Hours

Unit-II:
SWITCHING: Definition, Circuit switched networks, Data gram Networks, Virtual circuit networks,
Data Link Layer:ERROR DETECTION AND CORRECTION: cyclic codes, Checksum. DATA LINK CONTROL: Protocols for Noiseless channels, Noisy channels
Network Layer: Logical Addressing: IPv4 Addresses: Address Space, Notation, Classful Addressing, Classless Addressing, IPv6 Addresses: Structure, Address Space

12 Hours

Unit-III:
IPv4, IPv6, Transition from IPv4 to IPv6, ARP, RARP, ICMP, and IGMP. Delivery, Forwarding & Routing: Delivery, Forwarding, Routing Table, Routing protocols: Distance vector routing [RIP Description], Link state routing [OSPF Description], Path vector routing [BGP Description], Multicast routing protocols: unicast, multicast and broadcast [Only definitions with diagrams], MOSPF, CBT

14 Hours

Unit- IV
Application Layer: Domain Name System: Name Space, Domain Name Space, DNS In The Internet, Resolution, DNS Messages, Dynamic Domain Name System, Remote Logging, Electronic Mail and File Transfer.

14 Hours

TEXT BOOK:
[Unit-I: Chapters 1, 2 and 7, Unit-II: Chapters 8, 10, 11 and 19, Unit-III: Chapters 20, 21, and 22, Unit-IV: Chapters 23, 24, 25 and 26]

Note: Refer only the mentioned topics in the above units w.r.t the chapters given.

REFERENCES BOOKS:
UIS604C: UNIX SYSTEM PROGRAMMING
4 CREDITS (4-0-0)

UNIT I

1. INTRODUCTION 4 hrs
UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics
File Types, The UNIX and POSIX File System, The UNIX and POSIX File Attributes, Inodes in UNIX System v, UNIX Kernel Support for Files, Hard and Symbolic links

2. UNIX and POSIX API’S 9 hrs
General File API’s, File and record locking, Directory file API’s.

UNIT II

3. UNIX PROCESSES 4 hrs
Unix Kernel Support for Processes, Process API’s, Process Attributes, Changing Process Attributes,

4. UNIX SIGNALS 9 hrs
The Unix Kernel Support for Signals, signal, Signal mask, sigaction, the SIGCHLD Signal and the waitpid function, The sigsetjmp and siglongjmp Functions, kill alarm, Interval Timers and POSIX.1 Timers

UNIT III

5. INTER-PROCESS COMMUNICATION 13 hrs
POSIX.1b IPC methods, The UNIX system V IPC methods, Unix system V messages, POSIX.1b messages, Unix system V semaphores, Unix system V shared memory, Memory mapped I/O.

UNIT IV

6. SOCKETS and TL1 13 hrs
Sockets, A stream socket example, Client –server message handling example, TL1, TL1 class, Client-server message example, Datagram example

TEXT BOOK:

REFERENCE BOOK:
1. W. Richard Stevens; Advanced Programming in the Unix Environment, Addison Wesley/PHI
UNIT I
Communication In The Workplace: Role of Communication in Business, Process of Human Communication. Feedback, elements, objectives, principles of communication, Importance of communication, barriers in communication
Communication In Organization: Formal & informal communication, verbal & non-verbal communication, oral & written communication, horizontal and vertical communication

UNIT II
Writing For The Effect: Business Etiquette and need for effect, Conversational Style, You-view Point
Listening: Introduction, meaning of listening, poor listening habits, types of listening, Effective and ineffective listening skills, Strategies for effective listening, payoffs of effective listening, barriers of effective listening

UNIT III
Written forms of communication: Letters: Business letters, memos, E-mails,
Reports: Objectives, Characteristics of a report, types of reports, importance of reports, Formats, Prewriting, Structure of reports, Writing the reports, Revising, editing and proof reading.

UNIT IV
Research paper, Dissertation Correctness of communication Common Errors in Usage, Punctuation and capitalization, words commonly misspelt.

Text Books

Reference Books
1. Write a C program that creates a child process to read commands from the standard input and execute them (a minimal implementation of a shell like program). You can assume that no arguments will be passed to the commands to be executed.

2. Write C program filecopy that asks for an input path and output path and then copies the input to output. Input and output are ordinary files. Make copy operation faster.

3. Write a C program print out a directory tree. Prompt the user and accept name of the starting directory, print the name of the starting directory, read the directory and ignore everything except directories, print the names of any directories encountered along with any directories that they contain, indent each level of directory two spaces.

4. Write a C program to implement stat() function.

5. Write a C program sv, that copies file to dir/file1, dir/file2……….etc except that when a target file is newer than its source file, no copy is made and a warning is printed. To avoid making multiple copies of linked files, sv does not allow '/'s in any of the source files names.

6. Write a C program to do the following: Using fork() create a child process. The child process prints its own process id and id of its parent and then exits. The parent process waits for its child to finish (by executing wait()) and prints its own process id and the id of its child process and then exits.

7. Write a C program will do the following: The parent process calls pipe to allocate a pipe device file. It then calls fork to create a child process. Both the parent and the child can access the pipe. The child process is designated as the sender of the message to the parent.

8. Write a C program to implement popen() and pclose() system calls.

9. Write a C program that will do the following: The process signal mask is set to SIGTERM signal. The process then defines signal handler for the SIGINT signal and also specifies that the SIGSEGV signal is to be blocked when the process is handling the SIGINT signal. The process then suspends its execution via the pause API. Use sigaction() API to handle the signal.

10. Write a C program to implement kill command using kill API.

11. Program to illustrate IPC.

12. Program to illustrate Sockets.
1. Create the tables for the following with suitable assumptions:

**Employee and Salary**

a. Create the above tables by properly specifying the primary keys and the foreign keys
b. Generate the queries to do the following:
   (i) To sum the salary of each employee and sort it in descending order on the sum of basic.
   (ii) To list the employees who earn less than the average salary.
   (iii) To list the names of employees and salary details, whose basic is less than the average salary.

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

   PERSON (Driver id #: string, Name: string, Address: string)
   CAR (Regno: string, Model: string, Year: int)
   ACCIDENT (Report-number: int, Accd-Date: date, Location: string)
   OWNS (Driver-id #: string, Regno:string)

   PARTICIPATED (Driver-id: string, Regno:string, Report-Number: int, Damage Amount: int)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Demonstrate how you
   a. Update the damage amount to 25000 for the car with a specific Regno in the ACCIDENT table with report number 12.
   b. Add a new accident to the database.
(iii) Find the total number of people who owned cars that were involved in accidents in 2008.
(iv) Find the number of accidents in which cars belonging to a specific model were involved.

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

   STUDENT (Regno: string, Name: string, Major: string, Bdate:date)
   COURSE (Course #:int, Cname:string, Dept:string)
   ENROLL (Regno:string, Course#:int, Sem:int, Marks:int)
   BOOK _ ADOPTION (Course#:int, Sem:int, Book-ISBN:int)
   TEXT (Book-ISBN:int, Book-Title:string, Publisher:string, Author:string)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Demonstrate how you add a new textbook to the database and make this book be adopted by some department.
(iii) Produce a list of textbooks (include Course #, Book-ISBN, Book-Title) in the alphabetical order for courses offered by the ‘CS’ department that use more than two books.
(iv) List any department that has all its adopted books published by a specific publisher.

4. A database management system is to be created for a study centre to keep track of the students and the assignments records of the students. The database records the date of submission of assignments, the date of evolution, the date of viva, the date of declaration of results, who evaluated the assignment, and the mark list description in which the assignment marks were sent to regional office. The database also maintains the details of the evaluators. Perform the following activities for the description as given above.
a) Design the database with suitable integrity constraints and create the database
b) Write the following queries using SQL:
   (i) Find the list of the students who have not submitted even a single assignment
   (ii) Find the details of the evaluators by whom average marks awarded are more than 70%
   (iii) Find the students who have passed more than 5 assignments
   (iv) Find the list of students who have not appeared in VIVA. Make and state suitable assumptions, if any.

5. Design a database for maintaining the details of shows and ticketing for the shows of multiples. New perform the following activities for the system:
a) Create the database
b) Write the following queries using SQL
   (i) Find the details of the movies whose shows are not yet full
   (ii) Find the details of the movies that had been screened at least one year earlier to
   (iii) Find the names of those that have an overall state of 60% of capacity on all days of screening

6. Consider the following database for a banking enterprise:
   BRANCH(Branch-name:string, Branch-City:string, Assets:real)
   ACCOUNT(Acno:int, Branch-Name:string, Balance:real)
   DEPOSITOR(Customer-Name:string, Accno:int)
   CUSTOMER(Customer-Name:string, Customer-Street:string, Customercity:string)
   LOAN(Loan-Number:int, Branch-Name:string, Amount:real)
   BORROWER(Customer-Name:string, Loan-Number:int)

   (i) Create the above tables by properly specifying the primary keys and the foreign keys
   (ii) Find all the customers who have at least two accounts at the Main branch.
   (iii) Find all the customers who have an account at all the branches located in a specific city.
   (iv) Demonstrate how you delete tuples in ACCOUNT relation at every branch located in a specific city.
   (v) Find all loan numbers for loans made at the specific branch with loan amounts greater than Rs1200.
   (vi) Find all loan numbers for loans with loan amounts between Rs 90,000 and Rs100000.

Instructions:
1. The exercises are to be solved in an RDBMS environment like Oracle.
2. Enter at least five tuples for each relation.
3. Create suitable front end for querying and displaying the results.
4. Suitable tuples have to be entered so that queries are executed correctly.
5. Front end may be created using either VB or any other similar tool.
6. Generate suitable reports.
7. The student need not create the front end in the examination. The results of the queries may be displayed directly.
8. Relevant queries other than the ones listed along with the exercises may also be asked in the examination.
9. Questions must be asked based on lots.
10. Each group should be of maximum 4 members.
11. Group tasks will be assigned based on lots
12. 10(CIE) marks will be given to group task implementation based on demonstration.
13. In the SEE examinations the viva marks are based on knowledge of the subject and demonstration of group task.
Group Tasks (not limited to this list):

1. Library Management
2. Hotel Management
3. Attendance Report
4. Bus/Rail/Air Ticket Reservation Management
5. Employee Record System
6. Insurance Database
7. Order processing Database
8. Bank Database
9. Application for gold selling scheme.
11. Information management system.

General Remarks:
1) Lab schedule: 3hrs/week for each student (1 hr tutorial, 2 hrs- program execution).
2) Student should complete all the lab assignments.
3) One group task adequate per student batch (2 students/batch).
4) Each student should demonstrate his/her group work individually.
5) Evaluation CIE 50 marks:
   - Lab assignments : 30 marks
   - Group work : 10 marks
   - Lab CIE : 10 marks
Dynamic Documents with Java Script
1. The document must have a paragraph of text that describes your home. Choose at least three different phrases (3 to 6 words) of this paragraph and make them change font, font style, color and font size when the mouse cursor is placed over them. Each of the different phrases must change to different fonts, font styles, colors and font sizes.
2. The document must contain four short paragraphs of text stacked on top of each other with only enough of each showing so that the mouse cursor can also be placed over some part of them. When the cursor is placed over the exposed part of any paragraph it should raise to the top to become completely visible.
3. The document should have a small image of yourself, which must appear when the mouse button is clicked at the position of the mouse cursor, regardless of the position of the cursor at the time.
4. Develop and demonstrate, using Javascript script, 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.

XML
5. Design an XML document to store information about a student in an engineering college affiliated to VTU. 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.
6. (a) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.
   (b) Modify the XSLT style sheet to format the student information of the XML document of exercise 5 and use it to create a display of whole element.

PHP- Write, test and debug PHP scripts for the following specification
7. Parameter - A string of numbers separated by spaces
   Return Value – The first 4 digits number in the string; false if none
8. Parameter – A file variable of a file of text where the words are separated by spaces or colon
   Return value – The word that appears most common in the file.
9. 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.
10. Write the XHTML code to create the form with the following capabilities
    A text widget to collect the users name
    Four check boxes, one each for the following items
        • Four 100 watt light bulbs for Rs. 20=39
        • Eight 100 watt light bulbs for Rs 40=20
        • Four 100 watt long life light bulbs for Rs. 30=95
        • Eight 100 watt long life light bulbs for Rs 70=49
    A collection of 3 radio buttons that are labeled as follows
        • Visa
        • Master Card
        • Discover
11. Write a PHP script that computes the total cost of the ordered light bulbs from the exercise 10 after adding 13.5% VAT. The program must inform the buyer of exactly what was ordered in table.

12. Write a XHTML code to create a form that collects favorite popular songs, including the name of the song, the composer and the performing artist or group. This document must call one PHP script when the form is submitted and another to request a current list of survey results.

13. Write a XHTML code to provide a form that collects names and telephone numbers. The phone numbers must be in the format ddd-ddd-dddd. Write a PHP script that checks the submitted telephone number to be sure that it confirms to the required format and then returns a response that indicates whether the number was correct.

14. Accept from the user name, phone no, mail-ed, store in database. Retrieve same information from database using a separate PHP script.

15. Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.