SCHEME OF TEACHING AND EXAMINATION

B.E. (ISE) V SEMESTER

Sl No	Subject Code	Subject	Credits	Hours/Week			Examination Marks		
				Lecture	Tutorial	Practical	CIE	SEE	Total
1	UIS501C	System Software	04	4	0	0	50	50	100
2	UIS502C	Operating System	04	4	0	0	50	50	100
3	UIS503C	Database Management Systems	04	4	0	0	50	50	100
4	UIS504C	Data Communication	03	3	0	0	50	50	100
5		Elective – I	03	3	0	0	50	50	100
6		Open Elective – I	03	3	0	0	50	50	100
7	UIS505L	System Software Laboratory	1.5	0	1	2	50	50	100
8	UIS506L	Microprocessor Laboratory	1.5	0	1	2	50	50	100
9	UIS507L	Database Applications Laboratory	1.5	0	1	2	50	50	100
		Total	**25.5**	**21**	**3**	**6**	**450**	**450**	**900**

SCHEME OF TEACHING AND EXAMINATION

B.E. (ISE) VI SEMESTER

Sl. No.	Subject Code	Subject	Credits	Hours/Week			Examination Marks		
				Lecture	Tutorial	Practical	CIE	SEE	Total
1	UIS601H	Engineering Management.	04	4	0	0	50	50	100
2	UIS602C	Software Engineering.	04	4	0	0	50	50	100
3	UIS603C	Computer Networks.	04	4	0	0	50	50	100
4	UIS604C	Unix System Programming.	04	4	0	0	50	50	100
5	UIS605H	Professional Communication And Technical Writing.	02	2	0	0	50	50	100
6		Elective-II	03	3	0	0	50	50	100
7		Open Elective – II	03	3	0	0	50	50	100
8	UIS606L	Unix System Programming Laboratory	1.5	0	1	2	50	50	100
9	UIS607L	Operating System Laboratory.	1.5	0	1	2	50	50	100
		Total	**27**	**24**	**2**	**4**	**450**	**450**	**900**

V SEMESTER:

UIS501C: SYSTEMS SOFTWARE
CREDITS (4 – 0 – 0)
UNIT - 1

MACHINE ARCHITECTURE: Introduction, System Software and Machine Architecture, simplified Instructional Computer (SIC) – SIC Machine Architecture, SIC/XE Machine Architecture, SIC Programming Examples. RISC processor – Ultra SPARK Architecture, CISC processor – Pentium Pro Architecture

05 Hours

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

Machine Independent Assembler Features – Literals, Symbol-Defining Statements, Expression, Program Blocks, Control Sections and Programming Linking, Assembler Design Options - One-Pass Assembler, Multi-Pass Assembler.

08 Hours

UNIT - 2

LOADERS AND LINKERS: Basic Loader Functions - Design of an Absolute Loader, A Simple Bootstrap Loader, Machine-Dependent Loader Features – Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader; Machine-Independent Loader Features - Automatic Library Search, Loader Options, Loader Design Options - Linkage Editor, Dynamic Linking, Bootstrap Loaders.

08 Hours

MACRO PROCESSOR: Basic Macro Processor Functions – Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Machine-Independent Macro Processor Features - Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Keyword Macro Parameters, Macro Processor Design Options – Recursive Macro Expansion, General-Purpose Macro Processors, Macro Processing Within Language Translators.

05 Hours

UNIT - 3

COMPILERS: Basic Compiler Function – Grammars, Lexical Analysis, Syntactic Analysis, Code Generation, Machine Dependent Compiler Features: Intermediate Form of the Program, Machine-Dependent Code Optimization

Machine Independent Compiler Features: Structured Variables, Machine-Independent Code Optimization, Storage Allocation, Block-Structured Languages, Compiler Design Options: Division into Passes, Interpreters, P-Code Compilers, Compiler-Compilers.

13 Hours

UNIT - 4

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

Using YACC - Grammars, Recursive Rules, Shift/Reduce Parsing, What YACC Cannot Parse, A YACC Parser – The Definition Section, The Rules Section, Symbol Values and Actions, The

LEXER, Compiling and Running a Simple Parser, Arithmetic Expressions and Ambiguity, Variables and Typed Tokens.

09 Hours

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.

04 Hours

TEXT BOOKS:

1. **System Software** - Leland. L. Beck, 3rd Edition, Addison-Wesley,1997.
2. **Lex and Yacc** - John. R. Levine, Mason and Doug Brown, O'Reilly,SPD, 1998.

REFERENCE BOOK:

1. **System Programming and Operating Systems** – D.M.Dhamdhere,2nd Edition, Tata McGraw - Hill, 1999.

UIS502C: OPERATING SYSTEM
4 CREDITS (4-0-0)

UNIT 1

INTRODUCTION TO OPERATING SYSTEMS, SYSTEM STRUCTURES

Role of Operating systems: user view, system view; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and security; Distributed system; Special purpose systems; Computing environments. Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot. Process management: Process concept; Process scheduling; Operations on processes; Interprocess communication. Multi-Threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling; Thread scheduling. 13 Hrs

UNIT 2

PROCESS SYNCHRONIZATION

Synchronization: The Critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

13 Hrs

UNIT 3

MEMORY MANAGEMENT

Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Copy-on write; Page replacement; Allocation of frames; Thrashing, File system: File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection. Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.

13 Hrs

UNIT 4

SECONDARY STORAGE STRUCTURES, PROTECTION

Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix , Implementation of access matrix, Access control, Revocation of access rights, Capability-Based systems.

Case study: Linux operating system: Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory management; File systems, Input and output; Interprocess communication. 13 Hrs

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin , Greg Gagne: Operating System Principles, 7th edition, Wiley-India, 2006.

Reference Books:

1. D.M Dhamdhere: Operating systems - A concept based Approach, 2nd Edition, Tata McGraw-Hill, 2002.
2. P.C.P. Bhatt: Operating Systems, 2nd Edition, PHI, 2006.
3. Harvey M Deital: Operating systems, 3rd Edition, Addison Wesley, 1990.

UIS503C: DATABASE MANAGEMENT SYSTEMS
4 CREDITS (4:0:0)

UNIT – 1 (12hours)

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. (6)

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. (6)

UNIT – 2 (14hours)

RELATIONAL MODEL AND RELATIONAL DATABASE CONSTRAINTS: Relational model concepts; Relational model constraints and Relational database schemas; Update operations, Transaction and dealing with constraint violations. (4)

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. (5)

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; (5)

UNIT – 3 (13hours)

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 (6)

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. (7)

UNIT – 4 (13hours)

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; (8)

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

(5)

Text book :

1. Remez Elmasri & Shamkant B. Navathe "Fundamentals of Database Systems", 5th Edition, Pearson Education;

Reference book:

1. Ramakrishanan Gehrke " Database Management Systems", 3rd edition, McGraw-Hill Higher Education;
2. C. J. Date, "An Introduction to Data base systems", Addision Wesley, 4th edition.

UIS504C: DATA COMMUNICATIONS
3 CREDITS (3-0-0)

UNIT-I

Introduction:

Data Communications: Components, Data representations, Data flow, Networks: Network Criteria, physical structures, network models, categories of Networks, The Internet**:** Brief history, the internet today, Protocols and Standards: protocols, standards, standard organizations, internet standards, Layered tasks: sender, receiver and carrier The OSI Model: layered architecture, peer to peer processes, encapsulation, Layers in the OSI model, TCP / IP Protocol Suite: physical, data link, transport and application layer, Addressing: physical, logical and port addresses.

8 Hrs

Physical Layer and Media:

Digital Transmissions:

Digital-to-Digital conversion: line coding, line coding schemes, block coding, scrambling. Analog-to-Digital conversion: PCM, DM. Transmission modes: parallel and serial transmissions.

Analog Transmissions:

Digital - to - Analog conversion; ASK, FSK, PSK, QAM, Analog - to - Analog conversion: AM, FM, PM, Multiplexing and Demultiplexing: Frequency Division, Wavelength, time division

5 hrs

UNIT-II

TRANSMISSION MEDIA: Twisted pair cable, Coaxial cable, Fibre-Optic cable, Radio waves, Microwaves, Infrared. ERROR DETECTION AND CORRECTION: Introduction to error detection / correction, Block coding, linear block codes, cyclic codes, Checksum

4 Hrs

SWITCHING: Circuit switched networks, Data gram Networks, Virtual circuit networks, structure of a switch

5 Hrs

UNIT-III

DATA LINK CONTROL: Framing: fixed and variable sized framing, Flow and Error control, Protocols, Noiseless channels, Noisy channels, HDLC, Point-to-point Protocol: framing, transition phases.

5 Hrs

MULTIPLE ACCESSES: Random Access: CSMA, CSMA/CD, CSMA/CA, Controlled Access: reservation, polling, token poling, Canalizations: FDMA, TDMA, CDMA

4Hrs

UNIT-IV

Wired LANs: Ethernet

IEEE standards: Data link layer**,** physical layer, Standard Ethernet: MAC Sub layer, physical layer, changes in the standards: bridged Ethernet, switched Ethernet, full duplex Ethernet, Fast and Gigabit Ethernet

5 Hrs

WIRELESS LANS AND CONNECTION OF LANS

IEE 802.11: Archicture, MAC sub layer, Addressing mechanism, physical layer. Bluetooth. Connecting devices, Cellular telephony

6 Hrs

TEXT BOOK:

1. **Data Communications and Networking** – Behrouz A. Forouzan, 4th Edition, Tata McGraw-Hill, 2006.

REFERENCE BOOKS:

1. **Communication Networks: Fundamental Concepts and Key Architectures -** Alberto Leon, Garcia and Indra Widjaja, 3rd Edition, Tata McGraw- Hill, 2004.
2. **Data and Computer Communication,** William Stallings, 8th Edition, Pearson Education, 2007.
3. **Computer Networks: A Systems Approach -** Larry L. Peterson and Bruce S. David, 4th Edition, Elsevier, 2007.
4. **Introduction to Data Communications and Networking** – Wayne Tomasi, Pearson
2. Education, 2005.
5. **Computer and Communication Networks** – Nader F. Mir, Pearson Education, 2007.

UIS505L: SYSTEMS SOFTWARE LABORATORY
1.5 CREDITS (0 – 1- 2)

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 to implement pass1 functions of a 2-pass assembler.
3) Write a C program to generate machine code for instructions belonging to any one of the formats given below;
   F1: label opcode
   F2: label opcode r1 r2
   F3 : label opcode displacement
4) Write a c-program to implement pass1 functions of a 2 pass linking loader given an object file containing 2 control sections as input.
5) 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

PART B

Execution of the following programs using LEX:

1) Program to count the number of characters, words, spaces and lines in a given input file.
2) Program to count the numbers of comment lines in a given C program. Also eliminate them and copy that program into separate file.
3) Program to recognize a valid arithmetic expression and identify the identifiers and operators present. Print them separately.
4) Program to recognize whether a given sentence is simple or compound.
5) Program to recognize and count the number of identifiers in a given i/p file.

Execution of the following programs using YACC:

1) Program to test the validity of a simple expression involving operators +, -, * and /.
2) Program to evaluate an 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 recognize the grammar ($a^n b$, $n \geq 10$).
5) Program to recognize the grammer ($a^n b^n$,$n \geq 0$)

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.

UIS506L: MICROPROCESSOR LAB
1.5 CREDITS (0-1-2)

Part A: Assembly Language Assignments.

1. Write an assembly language program to simulate 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 any requests arrive while elevator is moving up or coming down they should be ignored.

UIS507L: DATABASE APPLICATIONS LABORATORY
1.5 CREDITS (0-1-2)

1. 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.

2. Consider the following relations for an order processing database application in a company:
CUSTOMER (Cust #: int , Cname: string, City: string)
ORDER (Order #: int, Odate: date, cust #: int, Ord-Amt: int)
ORDER – ITEM (Order #: int, item #: int, Qty: int)
ITEM (Item #: int, Unit price: int)
SHIPMENT (Order #: int, Warehouse#: int, Ship-Date: date)
WAREHOUSE (Warehouse #: int, City: string)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Produce a listing: **CUSTNAME, #oforders, AVG_ORDER_AMT**, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.
(iii) List the order# for orders that were shipped from *all* the warehouses that the company has in a specific city.
(iv) Demonstrate the deletion of an item from the **ITEM** table and demonstrate a method of handling the rows in the **ORDER_ITEM** table that contain this particular item.

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. The following tables are maintained by a book dealer:
AUTHOR (Author-id:int, Name:string, City:string, Country:string)
PUBLISHER (Publisher-id:int, Name:string, City:string, Country:string)
CATALOG (Book-id:int, Title:string, Author-id:int, Publisher-id:int, Category-id:int, Year:int, Price:int)
CATEGORY (Category-id:int, Description:string)
ORDER-DETAILS (Order-no:int, Book-id:int, Quantity:int)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Enter at least five tuples for each relation.
(iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
(iv) Find the author of the book, which has maximum sales.
(v) Demonstrate how you increase the price of books published by a specific publisher by 10%.

5. Consider the following database for a banking enterprise:
BRANCH(Branch-name:string, Branch-City:string, Assets:real)
ACCOUNT(Accno:int, Branch-Name:string, Balance:real)
DEPOSITOR(Customer-Name:string, Accno:int)
CUSTOMER(Customer-Name:string, Customer-Street:string, 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) Enter at least five tuples for each relation
(iii) Find all the customers who have at least two accounts at the Main branch.
(iv) Find all the customers who have an account at all the branches located in a specific city.
(v) Demonstrate how you delete tuples in ACCOUNT relation at every branch located in a specific city.

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.

Group Tasks:

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

VI SEMESTER:

UIS601H: ENGINEERING MANAGEMENT
4 CREDITS (4-0-0)

UNIT I

Engineering and Management: Introduction, What is Engineering, How is Management defined, Management levels, Who are the managers, Managerial Skills, Managerial roles, Function of managers, Process of the Management, What is Engineering Management, Structure of the Management in Engineering, Engineering Management Education program, The Historical background of Management, Scientific Management, Administrative Management, Behavioral Management.

13 Hours

UNIT II

Planning , Forecasting and Decision Making: Introduction, Planning process model, Strategic planning, Strategic planning process, Management by Objectives, Process of MBO, Planning concepts, Forecasting-Quantitative and Qualitative methods, Invention and Innovation , Nature of the Decision making, Types of Decision, Tools for Decision making.

Organizing and some Human Aspects of Organizing: Introduction, Nature of Organizing, Traditional Organizational theory, Span of control, factors of determining effective spans, Woodward and Aston studies, Human Resource planning, Selection and its process, Authority and Power, Delegation, Committees and Meetings.

13 Hours

UNIT III

Directing: Introduction, Motivation, Theory X and Theory Y, Content and process theories, Leadership- Types and nature of leadership, People/Task Matrix approaches, Leadership traits.

Controlling: Introduction, Process of Control, Three perspectives on timing of control, characteristics of effective control systems, Financial and Non-financial controls.

12 Hours

UNIT IV

Technology Management: Introduction, Product and Technology life cycles, Nature of Research Development, Selecting R&D projects, Quantitative approaches, Patents and Copyrights, Creativity, Creative process.

Managing Engineering Design: Introduction, Design phases, Concurrent Engineering, Control systems in design, Reliability and Risk, Simple reliability models, Materials Management, Production planning and control, Plant Location and Layout.

12 Hours

TEXT BOOKS:

1. Managing Engineering and Technology, Third Edition, Daniel L. Babcock and Lucy C. Morse, Prentice-Hall of India Private Limited, 2005, New Delhi.
2. Essentials of Management, Harold Koontz and Heinz Weihrich, Tata McGraw-Hill 1998.

REFERENCE BOOKS:

1. Management, 6th Edition, Don Hellrigel, John W.Slocum Jr. Addison-Wesley Publishing Company, 1991, Singapore.
2. Management, James A. F. Stonner, Second edition, Pretince-Hall of India Private Limited, New Delhi.
3. Engineering and Technology Management, V.S. Bagad, Technical Publications, Pune.
4. Essentials of Management, Joseph L. Massie, Fourth Edition, Prentice-Hall of India, Pearson, 2003.

UIS602C: SOFTWARE ENGINEERING
4 CREDITS (4-0-0)

UNIT – I **13 hours**

Introduction: Concepts of Software Engineering, Professional and ethical responsibilities. (1 hour)
Socio-Technical systems: Emergent system properties, systems engineering, organizations, people and computer systems, legacy systems. (3 hours)
Critical systems: A simple safety critical system, system dependability, Availability and reliability, safety, security. (2 hours)
Software Processes: software process models, process iteration, process activities, rational unifies process, computer aided software engineering. (7 hours)

UNIT – II **13 hours**

Software requirements: Functional and non functional requirements, user requirements, system requirements, the software requirements document. (4 hours)
Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management. (5 hours)
System Models: Context based models, behavioral models, data models, object models, structured methods. (4 hours)

UNIT – III **13 hours**

Architectural design: Architectural design decisions, system organization, modular decomposition styles, control styles, reference architectures. (4 hours)
Object oriented design: objects and object classes, an object oriented design process, design evolution. (3 hours)
User interface design: Design issues, UI design process, user analysis, user interface prototyping, interface evaluation. (3 hours)
Rapid software development: agile methods, extreme programming, rapid application development, software prototyping.
(3 hours)

UNIT – IV **13 hours**

Verification and validation: Planning verification and validation, software inspections, automated static analysis, verification and formal methods. (4 hours.)
Software testing: system testing, component testing, test case design, test automation. (3 hours.)
Managing people: selecting staff, motivating people, managing groups, People capability maturity model. (2 hours)
Software cost estimation: software productivity, estimation techniques, algorithmic cost modeling, project duration and staffing. (2 hours)
Quality Management: Process and product quality, quality assurance and standards, CMMI process framework. (2 hours)

Text Book:

1. Ian Somerville, Software Engineering, 7th edition, Person Education.

Reference Books:

2. Pressman R.S, “Software Engineering A Practitioners Approach”, MGH New Delhi.
3. Jalote P, “An Intergrated Approach to Software Engineering”, Narosa New Delhi.

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

UNIT-I

ATM Networks:
Design Goals, Problems, Architecture, Switching, ATM Layers, ATM LANs: ATM LAN Architecture, LAN Emulation.

7 Hrs

Network Layer: Logical Addressing:
IPv4 Addresses: Address Space, Notation, Classful Addressing, Classless Addressing, IPv6 Addresses: Structure, Address Space

5 Hrs

UNIT-II

Network Layer: Internet Protocol:
Internetworking, IPv4, IPv6, Transition from IPv4 to IPv6. Address Mapping, Error Reporting & Multicasting: Address Mapping, ICMP, IGMP. Delivery, Forwarding & Routing: Delivery, Forwarding, Unicast Routing Protocols: Optimization, Intra- and Interdomain Routing, Distance vector routing, Link state routing, Multicast routing protocols: unicast, multicast and broadcast, multicast routing, routing protocols: MOSPF, DVMRP, Flooding, RPF, RPB, CBT, PIM.

14 Hrs

UNIT-III

Transport Layer:
Process-to-Process Delivery: UDP, TCP, SCTP. Congestion Control

6 Hrs

Quality of Service: Flow characteristics, Flow classes, Techniques to improve QoS, Integrated Services, Differentiated Services, QoS in Switched Networks.

6 Hrs

UNIT-IV

Application Layer:
Domain Name System: Name Space, Domain Name Space, Distribution of Name Space, DNS In The Internet, Resolution, DNS Messages, Types of records, Registers, Dynamic Domain Name System, Encapsulation. Remote Logging, Electronic Mail, File Transfer. WWW and HTTP: Architecture, Web Documents, HTTP
Network Management: Network Management System, SNMP.

14 Hrs

TEXT BOOKS:

1. **Data Communications and Networking** - Behrouz A. Forouzan, 4th Edition, Tata McGraw-Hill, 2006.

REFERENCES BOOKS:

1. **Communication Networks –Fundamental Concepts and Key Architectures** - Alberto Leon-Garcia and Indra Widjaja, 2nd Edition, Tata McGraw-Hill, 2004.
2. **Computer and Communication Networks** - Nader F. Mir, Pearson Education, 2007.
3. **Data and Computer Communication** - William Stallings, 8th Edition, Pearson Education, 2007.
4. **Computer Networks – A Systems Approach** - Larry L. Peterson and Bruce S. David, 4th Edition, Elsevier, 2007.
5. **Introduction to Data Communications and Networking –** Wayne Tomasi, Pearson Education, 2005.

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

UNIT I

1. **BACKGROUND AND SOME BASIC COMMANDS** **3 hrs**

Brief history and Salient features of a Unix System, POSIX and the Single UNIX specification, The UNIX Architecture, Locating Commands. Internal and External Commands, Flexibility of Commands Usage, **man**: Browsing and MANUAL pages On-line, Understanding the man Documentation.

cal: The calendar, **date**: Displaying and System Date, **echo**: Displaying a message**, printf:** An Alternative to echo, **bc**: The Calculator, **script**: Recording Your Session, **passwd**: Changing Your Password, **who**: Who Are Users?, **uname**: Knowing Your Machine's Characteristics, **tty:** Knowing Your Terminal, **stty**: Displaying and Setting Terminal Characteristics.

2. THE FILE SYSTEM AND SOME FILE HANDLING COMMANDS **6 hrs**

The File, Content of the file, The Parent and Child Relationship, The HOME Varible: The Home Directory, pwd: Checking Your Current Directory, cd: Changing the Current Directory, mkdir: Making Directories, rmdir: Removing Directories, Absolute Pathnames, Relative Pathnames, ls: Listing Directory Contents, The UNIX File System

cat: Displaying and Creating Files, cp: Copying a File, rm: Deleting Files, mv: Renaming Files, more: Paging Output, The lp Subsystem: Printing a Characters, od: Displaying Data in octal, FILE ATTRIBUTES: ls –l: Listing File attributes, The –d option: Listing Directory Attributes, listing inode number, listing hidden files, time associated with a file, listing timestamps, File Ownership, File Permissions, chmod: Changing File Permissions, Directory Permissions, Changing File Ownership, File System and Inodes, Hard Links, Symbolic Links and ln, The directory, umask: Default File and Directory Permissions, Modification and Access Times, find: Locating Files.

3. THE vi EDITOR **4hrs**

Vi Basics, Input Mode Entering and Replacing Text, Saving Text and Quoting- The ex Mode, Navigation, Editing Text, Undoing Last Editing Instructions (u and U), Repeating the Last Commands(.), Searching for a Pattern (/ and ?), Substitution-Search and Replace(:s), Customizing vi.

UNIT II

4. THE SHELL AND THE PROCESS **3 hrs**

The shell's Interpretive Cycle, Pattern Matching- The Wild-cards, Escaping and Quoting, Redirection: The Three standard Files, /dev/null and /dev/tty: Two Special Files, Pipes, tee: Creating a Tee, Command Substitution, Shell Variables.

Process Basics, ps: Process Status, System Processes (-e or –a), Mechanism of Process creation, Internal and External Commands, Running Jobs in Background, nice: Job execution with low priority, Killing Processes with Signals, Job Control, at: and batch: Execute Later, cron: Running Jobs Periodically, time: Timing Processes

5. SIMPLE FILTERS AND grep FAMILY OF COMMANDS 3 hrs

The Sample Data base, pr: Paging Files, head: Displaying the Beginning of a File, tail: Displaying the End of a File, cut: Splitting a File Vertically, paste: Pasting Files, sort: Ordering a File, Uniq; Locate Repeated and Non repeated Lines, tr: Translating Characters, An Example : Displaying a word count List, grep: Searching for a Pattern, Basic Regular Expressions- An Introduction, Extended Regular Expressions and egrep

6. SHELL PROGRAMMING 7 hrs

Environment Variables, Aliases (bash and ksh), Command History (bash and ksh), Shell Scripts, read and readonly commands, Using Command line Arguments, exit and exit status of the command, The Logical Operators && and ||- Conditional Execution. The if Conditional, Using test and [ ] to Evalutate Expressions, The case Conditional, expr: Computation and String Handling, $0: Calling a Script by Different names, while: Looping, for: Looping with a List, set and shift: Manipulating the Positional Parameters, The here Document (<<), trap: Interrupting a Program, Debugging Shell Scripts with set –x, export: Exporting Shell Variables, eval: Evaluating Twice.

UNIT III

7. INTRODUCTION 6 hrs

UNIX and ANSI Standards : The ANSI C Stantard, The ANSI/ISO C++ Standards, Differenece between ANSI C and C++, The POSIX Standard, The POSIX.1 FIPS Standard, The X/Open Standards.

UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics

File Types, The UNIX and POSIX File sytem, The UNIX and POSIX File Attributes, Inodes in UNIX System v, UNIX Kernel Support for Files, Hard and Symbolic links.

8. UNIX FILE API's 7 hrs

General File API's, File and record locking, Directory file API's, Device File API's, FIFO File API's, Symbolic Link File API's

UNIT IV

9. UNIX PROCESSES 6 hrs

Unix Kernel Support for Processes, Process API's, Process Attributes, Changing Process Attributes.

10. UNIX SIGNALS 7 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.

TEXT BOOKS:

2. 1 Sumitaba Das, Unix Concepts and Applications, Third edition Tata McGraw Hill 2007
3. 2..Terrence Chan: Unix System Programming Using C++, Printice Hall India, 1999

REFERENCE BOOKS

1. 1.Behrouz A. Forouzan and Richard F. Gilberg, Unix and Shell Programming, A Text Book, Thomson, Edition-2003.
2. 2.W. Richard Stevens; Advanced Programming in the Unix Environment, Addison Wesley/PHI
3. Brain W. Kernighan and Rob Pike, The Unix Programming Environment Pearson Education, Edition -2006
4. Ellie Quigley, Unix Shells By Examples, Pearson Edition, Fourth Edition

UIS605H: PROFESSIONAL COMMUNICATION AND TECHNICAL WRITING
2 CREDITS (2-0-0)

Objectives of the Course:

- To understand the concept and meaning of Business communication.
- To know the flow of communication in the organization.
- To know what is listening and its importance in communication
- To understand different forms of written communication.
- To understand Public speaking and oral reporting.

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,

06 hours

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,

06 hours

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.

07 hours

UNIT IV

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

7 hours

Text Books

1. 1. Lesikar and Fatley , Basics Business communication Skills for Empowering the Internet

Generation 10th edition,; Tata McGraw Hill edition, ISBN: 978-0-07-059975-8.

2. Meenakshi Raman and Sangeeta Sharma "Technical Communication Principles and practices", Oxford University Press, ISBN-13 978-0-19-566804-9.

3. Meenakshi Raman and Prakash Singh "Business Communication", Oxford University Press, ISBN-13: 978-0-19-567695-2.

Reference Books

4. M Ashraf Rizvi " Effective Technical Communication", Tata McGraw Hill Company Limited, ISBN: 978-0-07-059952-9.

5. P.D.Chaturvedi Mukesh Chaturvedi " Business Communication- Concepts, Cases and Application", Pearson Education, ISBN:81-317-0172-7.

6. Rajendra Pal and J S Khorahalli , Essential of Business Communication-, S Chand and Sons Publications.

7. Urmita Rai nad S,M Rai ,Business Communication-, Himalaya Publishing House.

8. Krishna Mohan and Meera Banerjee , "Developing Communication Skills",
1. McMillan India Ltd.

9. Asha Kaul , "Business Communication", Prentice Hall India Pvt Ltd.

UIS606L: UNIX SYSTEM PROGRAMMING LAB
1.5 CREDITS (0-1-2)

SHELL PROGRAMS

1. Write a shell script called greetme that will do the following: Greet the user, print the date and the time, print a calendar for this month, print the name of your machine, print the name and release of this operating system, print a list of all files in your parent directory, print all the processes root is running, print the value of the TERM, PATH, and HOME variables, print your disk usage, print your group id.

2. Write a shell script called info that will do the following: ask the user's full name-first name, last name and middle name, greet the user by his or her first name, ask the user's year of birth and calculate his or her age, ask the user's login name and print his or her user id, tell the user his or her home directory, show the user processes he or she is running, tell the user the day of the week, and the current time.

3. Create a text file called datafile. Each entry consists of fields separated by colons. The fields are as follows: first name and last name, phone number, address, birth date, salary. Create a script called lookup that will do the following: sort datafile by last names, show the user the contents of datafile, tell the user the number of entries in the file

4. Write a shell script to find any files in the root partition that have not been modified within the past n days and are larger than 20 blocks(512 byte blocks)

5. Write a shell script that accepts valid file names as command line arguments and for each of the arguments, prints the type of the file (regular file, directory file character device file, block device file, symbolic link file etc and also print permission on file.

6. Write a shell script to find and display all the links of a file specified as the first argument to the script. The second argument, which is optional can be used to specify the directory in which the search is to begin. If this second argument is not present, the search is to begin in current working directory. In either case the starting directory as well as its subdirectories at all levels must be searched. The script need not include any error checking.

7. Write a shell script to provide nicer interface to cal command without changing cal itself. It should recognize month by name. With two arguments, it should behave just as the old cal does, except for converting month names into numbers. Given one argument, it should print the month or year's calendar as appropriate, and given zero arguments, it should print the current month's calendar.

8. Write a shell script to report which file correspond to a command is executed.

9. Write a shell script to implement getopts statement, your script should understand following command line arguments and called this script Q14 –c –d –e where options work as
   -c clear the screen

-d show list of files in current working directory

10. Write a shell script called sayHello, put this script into your startup file called .bash_profile, the script should run as soon as you logon to system, and it print any one of the following message in infobox using dialog utility, if installed in your system, if dialog utility is not installed then use echo statement to print message: Good morning, Good afternoon, Good evening, according to system time

Unix Sytem programming:

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 diir/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 aC 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 that 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 vhild can access the pipe. The child process is designated as the sender of the message to the parent.
8. Write a C program that will do the following: The program is invoked with a shell command only (i.e., 1st argument), the program will call popen to executed the command and print the exec'ed command's standard output data to the console. If however the program is invoked a shell command and second argument the program will execute the shell command via popen, and supply the 2nd argument as data to the standard input part of the exec'ed command.

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

UIS607L: OPERATING SYSTEM LABORATORY
1.5 CREDITS (0-1-2)

Note: i. Assignments are to be implemented in Linux/Windows Environment

ii Instructor is expected to supply needed case instances parameters for Implementations.

iii. Marks: 50 Marks CIE + 50 SEE

Assignment 1:
Write a C/C++ program to implement communication between processes using Pipe mechanism. Your program should create a pipe to which sending process will write and the receiving process will read from the pipe.

Assignment 2:
Write a C/C++ program to compute average waiting time and average turnaround time for FCFS algorithm. The program should accept arrival time and burst time as input.

Assignment 3:
Write a C/C++ program to compute average waiting time and average turnaround time for SJF algorithm. The program should accept arrival time and burst time as input.

Assignment 4:
Write a C/C++ program to compute average waiting time and average turnaround time for priority scheduling algorithm. The program should accept arrival time, burst time and priority algorithm as input.

Assignment 5 :
Write a C/C++ program to compute average waiting time and average turnaround time for Round robin scheduling algorithm. The program should accept arrival time and burst time as input. Assume suitable time quantum as input.

Assignment 6:
Write a C/C++ program to illustrate multithreading concept. Your program should create n threads(ex:5) each thread displaying some message.

Assignment 7:
Write a program to implement Petersons solution to solve critical section problem..

Assignment 8:
Write a solution to the bounded buffer problem in C/C++ .

Assignment 9:
Write a program to detect whether the system is in safe state, the program should accept allocation,max and available matrices. Generate the need matrix.

Assignment 10:
Write a program to implement Contiguous memory allocation scheme for a system consisting of several processes.

Assignment 11:
Write a C/C++ program that implements FIFO page replacement algorithm.

Assignment 12:
Write a C/C++ program to implement LRU page replacement algorithm

Question Paper Pattern for SEE (for every theory subject):
Total of eight questions, two questions from each unit carrying twenty marks each, are to be set from each unit for SEE examination**.** Each FULL question should have a maximum of four sub questions. A student is expected to answer FIVE full questions choosing at least one FULL question from each unit.

Evaluation scheme for theory subjects:
A student will be evaluated in the subject through Continuous Internal Evaluation (CIE) and Semester End Examinations (SEE).

The CIE is for 50 marks and consists of 3 tests of 15 marks each and an assignment for 5 marks. The CIE tests are conducted for 1 hr for 30 marks, and marks obtained are scaled down to 15.

The SEE for 50 marks is conducted as 3 hrs exam for 100 marks, then the marks obtained are scaled down to 50.

Evaluation scheme for laboratory subjects:
A student will be evaluated in the subject through Continuous Internal Evaluation (CIE) and Semester End Examinations (SEE).

The CIE is for 50 marks. The students are evaluated for 30 marks for the conduct of the laboratory assignments and journal writing. The student is evaluated trough an internal lab test for the remaining 20 marks.

The SEE for 50 marks is through a lab exam of 3 hrs duration. In the lab exam student is evaluated as following:

a. 25% of total marks for initial write-up and overall submission: 12.5 marks.
b. 50% of total marks for conduct/completion of the task: 25 marks.
c. 25% of total marks for viva-voce on the lab/subject: 12.5 marks.

The student is awarded ‘S’ to ‘F’ grade based on his/her performance in CIE and SEE taken together.