V SEMESTER

SOFTWARE ENGINEERING

Subject Code: 10IS51	**I.A. Marks : 25**
Hours/Week : 04	**Exam Hours: 03**
Total Hours : 52	**Exam Marks: 100**

PART – A

UNIT – 1   **6 Hours**
Overview: Introduction: FAQ's about software engineering, Professional and ethical responsibility.
Socio-Technical systems: Emergent system properties; Systems engineering; Organizations, people and computer systems; Legacy systems.

UNIT – 2   **6 Hours**
Critical Systems, Software Processes: Critical Systems: A simple safety-critical system; System dependability; Availability and reliability.
Software Processes: Models, Process iteration, Process activities; The Rational Unified Process; Computer Aided Software Engineering.

UNIT – 3   **7 Hours**
Requirements: Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; Interface specification; The software requirements document.
Requirements Engineering Processes: Feasibility studies; Requirements elicitation and analysis; Requirements validation; Requirements management.

UNIT – 4   **7 Hours**
System models, Project Management: System Models: Context models; Behavioral models; Data models; Object models; Structured methods.
Project Management: Management activities; Project planning; Project scheduling; Risk management

PART - B

UNIT – 5   **7 Hours**
Software Design: Architectural Design: Architectural design decisions; System organization; Modular decomposition styles; Control styles.

Object-Oriented design: Objects and Object Classes; An Object-Oriented design process; Design evolution.

UNIT – 6   **6 Hours**
Development: Rapid Software Development: Agile methods; Extreme programming; Rapid application development.
Software Evolution: Program evolution dynamics; Software maintenance; Evolution processes; Legacy system evolution.

UNIT – 7   **7 Hours**
Verification and Validation: Verification and Validation: Planning; Software inspections; Automated static analysis; Verification and formal methods.
Software testing: System testing; Component testing; Test case design; Test automation.

UNIT – 8   **6 Hours**
Management: Managing People: Selecting staff; Motivating people; Managing people; The People Capability Maturity Model.
Software Cost Estimation: Productivity; Estimation techniques; Algorithmic cost modeling, Project duration and staffing.

Text Books:
1. Ian Sommerville: Software Engineering, 8th Edition, Pearson Education, 2007.
   (Chapters-: 1, 2, 3, 4, 5, 6, 7, 8, 11, 14, 17, 21, 22, 23, 25, 26)

Reference Books:
1. Roger.S.Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill, 2007.
2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India, 2009.

SYSTEM SOFTWARE

Subject Code: 10CS52   **I.A. Marks   : 25**
Hours/Week : 04   **Exam Hours: 03**
Total Hours : 52   **Exam Marks: 100**

PART – A

UNIT – 1   **6 Hours**
Machine Architecture: Introduction, System Software and Machine Architecture, Simplified Instructional Computer (SIC) - SIC Machine Architecture, SIC/XE Machine Architecture, SIC Programming Examples.

33

UNIT – 2   **6 Hours**
Assemblers -1: Basic Assembler Function - A Simple SIC Assembler, Assembler Algorithm and Data Structures, Machine Dependent Assembler Features - Instruction Formats & Addressing Modes, Program Relocation.

UNIT – 3   **6 Hours**
Assemblers -2: Machine Independent Assembler Features – Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking, Assembler Design Operations - One-Pass Assembler, Multi-Pass Assembler, Implementation Examples - MASM Assembler.

UNIT – 4   **8 Hours**
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 Linkage, Bootstrap Loaders, Implementation Examples - MS-DOS Linker.

PART – B

UNIT – 5   **6 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

UNIT – 6   **8 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, Implementation Examples - MASM Macro Processor, ANSI C Macro Processor.

UNIT – 7   **6 Hours**
Lex and Yacc – 1: 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.

UNIT – 8   **6 Hours**
Lex and Yacc - 2: 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.

Text Books:
1. Leland.L.Beck: System Software, 3rd Edition, Pearson Education, 1997.
   (Chapters 1.1 to 1.3, 2 (except 2.5.2 and 2.5.3), 3 (except 3.5.2 and 3.5.3), 4 (except 4.4.3))
2. John.R.Levine, Tony Mason and Doug Brown: Lex and Yacc, O'Reilly, SPD, 1998.
   (Chapters 1, 2 (Page 2-42), 3 (Page 51-65))

Reference Books:
1. D.M.Dhamdhere: System Programming and Operating Systems, 2nd Edition, Tata McGraw - Hill, 1999.

OPERATING SYSTEMS

Subject Code: 10CS53   I.A. Marks   : 25
Hours/Week : 04   Exam Hours: 03
Total Hours : 52   Exam Marks: 100

PART – A

UNIT – 1   **6 Hours**
Introduction to Operating Systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and security; Distributed system; Special-purpose systems; Computing environments.Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot.

UNIT – 2   **7 Hours**
Process Management: Process concept; Process scheduling; Operations on processes; Inter-process communication. Multi-Threaded Programming: Overview; Multithreading models; Thread Libraries;   Threading issues. Process Scheduling: Basic concepts; Scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling; Thread scheduling.

UNIT – 3   **7 Hours**
Process Synchronization : Synchronization: The Critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

UNIT – 4   **6 Hours**
Deadlocks: Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

PART – B

UNIT – 5   **7 Hours**
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.

UNIT – 6   **7 Hours**
File System, Implementation of File System: File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection. Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management

UNIT – 7   **6 Hours**
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.

UNIT – 8   **6 Hours**
Case Study: The Linux Operating System: Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory management; File systems, Input and output; Inter-process communication.

Text Books:
1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating System Principles, 8th edition, Wiley India, 2009.
   (Listed topics only from Chapters 1 to 12, 17, 21)

Reference Books:
1. D.M Dhamdhere: Operating systems - A concept based Approach, 2nd Edition, Tata McGraw- Hill, 2002.
2. P.C.P. Bhatt: Introduction to Operating Systems: Concepts and Practice, 2nd Edition, PHI, 2008.
3. Harvey M Deital: Operating systems, 3rd Edition, Pearson Education, 1990.

DATABASE MANAGEMENT SYSTEMS

Subject Code: 10CS54   I.A. Marks : 25
Hours/Week : 04   Exam Hours: 03
Total Hours : 52   Exam Marks: 100

PART - A

UNIT – 1   **6 Hours**
Introduction: Introduction; An example; Characteristics of Database approach; Actors on the screen; Workers behind the scene; Advantages of using DBMS approach; A brief history of database applications; 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.

UNIT – 2   **6 Hours**
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 – 3   **8 Hours**
Relational Model and Relational Algebra : Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update

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

UNIT – 4   **6 Hours**
SQL – 1: SQL Data Definition and Data Types; Specifying basic constraints in SQL; Schema change statements in SQL; Basic queries in SQL; More complex SQL Queries.

PART - B
UNIT – 5   **6 Hours**
SQL – 2:   Insert, Delete and Update statements in SQL; Specifying constraints as Assertion and Trigger; Views (Virtual Tables) in SQL; Additional features of SQL; Database programming issues and techniques; Embedded SQL, Dynamic SQL; Database stored procedures and SQL / PSM.

UNIT – 6   **6 Hours**
Database Design – 1: 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

UNIT – 7   **6 Hours**
Database Design -2: 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 – 8   **8 Hours**
Transaction Management: The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of locking; Transaction support in SQL; Introduction to crash recovery; 2PL, Serializability and Recoverability; Lock Management; Introduction to ARIES; The log; Other recovery-related structures; The write-ahead log protocol; Checkpointing; Recovering from a System Crash; Media Recovery; Other approaches and interaction with concurrency control.

Text Books:
1. Elmasri and Navathe: Fundamentals of Database Systems, 5th Edition, Pearson Education, 2007.

(Chapters 1, 2, 3 except 3.8, 5, 6.1 to 6.5, 7.1, 8, 9.1, 9.2 except SQLJ, 9.4, 10)

2. Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw-Hill, 2003. (Chapters 16, 17.1, 17.2, 18)

Reference Books:
1. Silberschatz, Korth and Sudharshan: Data base System Concepts, 6th Edition, Mc-GrawHill, 2010.
2. C.J. Date, A. Kannan, S. Swamynatham: An Introduction to Database Systems, 8th Edition, Pearson Education, 2006.

COMPUTER NETWORKS - I

Subject Code: 10CS55   I.A. Marks : 25
Hours/Week : 04   Exam Hours: 03
Total Hours : 52   Exam Marks: 100

PART – A

UNIT - 1   **7 Hours**
Introduction: Data Communications, Networks, The Internet, Protocols & Standards, Layered Tasks,
The OSI model, Layers in OSI model, TCP/IP Protocol suite, Addressing

UNIT- 2   **7 Hours**
Physical Layer-1: Analog & Digital Signals, Transmission Impairment, Data Rate limits, Performance, Digital-digital conversion (Only Line coding: Polar, Bipolar and Manchester coding), Analog-to-digital conversion (only PCM), Transmission Modes, Digital-to-analog conversion

UNIT- 3   **6 Hours**
Physical Layer-2 and Switching: Multiplexing, Spread Spectrum, Introduction to switching, Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks

UNIT- 4   **6 Hours**
Data Link Layer-1: Error Detection & Correction: Introduction, Block coding, Linear block codes, Cyclic codes, Checksum.

PART - B

UNIT- 5   **6 Hours**
Data Link Layer-2: Framing, Flow and Error Control, Protocols, Noiseless
Channels, Noisy channels, HDLC, PPP (Framing, Transition phases only)

UNIT- 6   **7 Hours**
Multiple Access & Ethernet: Random access, Controlled Access,
Channelization, Ethernet: IEEE standards, Standard
 Ethernet, Changes in the standard, Fast Ethernet, Gigabit Ethernet

UNIT - 7   **6 Hours**
Wireless LANs and Cellular Networks: Introduction, IEEE 802.11,
Bluetooth, Connecting devices, Cellular Telephony

UNIT - 8:   **7 Hours**
Network Layer: Introduction, Logical addressing, IPv4 addresses, IPv6
addresses, Internetworking basics,
IPv4, IPv6, Comparison of IPv4 and IPv6 Headers.

Text Books:
1. Behrouz A. Forouzan,: Data Communication and Networking, 4th
Edition Tata McGraw-Hill, 2006.
(Chapters 1.1 to 1.4, 2.1 to 2.5, 3.1 To 3.6, 4.1 to 4.3, 5.1, 6.1, 6.2,   8.1
to 8.3, 10.1 to 10.5, 11.1 to 11.7, 12.1 to 12.3, 13.1 to 13.5,   14.1, 14.2,
15.1, 16.1, 19.1, 19.2, 20.1 to 20.3)

Reference Books:
1. Alberto Leon-Garcia and Indra Widjaja: Communication Networks -
Fundamental Concepts and Key architectures, 2nd Edition Tata
McGraw-Hill, 2004.
2. William Stallings: Data and Computer Communication, 8th Edition,
Pearson Education, 2007.
3. Larry L. Peterson and Bruce S. Davie: Computer Networks – A
Systems Approach, 4th Edition, Elsevier, 2007.
4. Nader F. Mir: Computer and Communication Networks, Pearson
Education, 2007.

FORMAL LANGUAGES AND AUTOMATA THEORY

Subject Code: 10CS56 **I.A. Marks : 25**
Hours/Week : 04   **Exam Hours: 03**
Total Hours : 52   **Exam Marks: 100**

PART - A

UNIT – 1   **7 Hours**
Introduction to Finite Automata: Introduction to Finite Automata; The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata

UNIT – 2   **7 Hours**
Finite Automata, Regular Expressions: An application of finite automata; Finite automata with Epsilon-transitions; Regular expressions; Finite Automata and Regular Expressions; Applications of Regular Expressions

UNIT – 3   **6 Hours**
Regular Languages, Properties of Regular Languages: Regular languages; Proving languages not to be regular languages; Closure properties of regular languages; Decision properties of regular languages; Equivalence and minimization of automata

UNIT – 4   **6 Hours**
Context-Free Grammars And Languages : Context –free grammars; Parse trees; Applications; Ambiguity in grammars and Languages   .

PART – B

UNIT – 5   **7 Hours**
Pushdown Automata: Definition of the Pushdown automata; the languages of a PDA; Equivalence of   PDA's and CFG's; Deterministic Pushdown Automata

UNIT – 6   **6 Hours**
Properties of Context-Free Languages: Normal forms for CFGs; The pumping lemma for CFGs; Closure properties of CFLs

UNIT – 7   **7 Hours**
Introduction To Turing Machine: Problems that Computers cannot solve; The turning machine; Programming techniques for Turning Machines;

Extensions to the basic Turning Machines; Turing Machine and Computers.

UNIT – 8   **6 Hours**
Undecidability: A Language that is not recursively enumerable; An Undecidable problem that is RE; Post's Correspondence problem; Other undecidable problems.

Text Books:
1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson Education, 2007.
   (Chapters: 1.1, 1.5, 2.2 to 2.5, 3.1 to 3.3, 4, 5, 6, 7, 8.1 to 8.4, 8.6, 9.1, 9.2, 9.4.1, 9.5)

Reference Books:
1. K.L.P. Mishra: Theory of Computer Science, Automata, Languages, and Computation, 3rd Edition, PHI Learning, 2009.
2. Raymond Greenlaw, H.James Hoover: Fundamentals of the Theory of Computation, Principles and Practice, Elsevier, 1998.
3. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.
4. Thomas A. Sudkamp: An Introduction to the Theory of Computer Science, Languages and Machines, 3rd Edition, Pearson Education, 2006.

DATABASE APPLICATIONS LABORATORY

Subject Code: 10CSL57   **I.A. Marks : 25**
Hours/Week : 03   **Exam Hours: 03**
Total Hours : 42   **Exam Marks: 50**

1. Consider the following relations:
   Student (*snum:* integer, *sname:* string, *major:* string, *level:* string, *age:* integer)
   Class (*name:* string, *meets at:* string, *room:* string, *d:* integer)
   Enrolled (*snum:* integer, *cname:* string)
   Faculty (f*id*: integer, *fname:* string, *deptid:* integer)
   The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level is a two character code with 4 different values (example: Junior: JR etc)
   Write the following queries in SQL. No duplicates should be printed in any of the answers.

i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Prof. Harshith
ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
iii. Find the names of all students who are enrolled in two classes that meet at the same time.
iv. Find the names of faculty members who teach in every room in which some class is taught.
v. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

2. The following relations keep track of airline flight information:
   Flights (*no:* integer, *from:* string, *to:* string, *distance:* integer, *Departs:* time, *arrives:* time, *price:* real)
   Aircraft (*aid:* integer, *aname:* string, *cruisingrange:* integer)
   Certified (*eid:* integer, *aid:* integer)
   Employees (*eid:* integer, *ename:* string, *salary:* integer)
   Note that the Employees relation describes pilots and other kinds of employees as well;   Every pilot is certified for some aircraft, and only pilots are certified to fly.
   Write each of the following queries in SQL.

   i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
   ii. For each pilot who is certified for more than three aircrafts, find the *eid* and the maximum *cruisingrange* of the aircraft for which she or he is certified.
   iii. Find the names of pilots whose *salary* is less than the price of the cheapest route from Bengaluru to Frankfurt.
   iv. For all aircraft with *cruisingrange* over 1000 Kms, .find the name of the aircraft and the average salary of all pilots certified for this aircraft.
   v. Find the names of pilots certified for some Boeing aircraft.
   vi. Find the *aid*s of all aircraft that can be used on routes from Bengaluru to New Delhi.

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. Enter at least five tuples for each relation.
iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
v. List any department that has *all* its adopted books published by a specific publisher.
vi. Generate suitable reports.
vii. Create suitable front end for querying and displaying the results.

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%.
vi. Generate suitable reports.
vii. Create suitable front end for querying and displaying the results.

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, customer-city: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 all account tuples at every branch located in a specific city.
vi. Generate suitable reports.
vii. Create suitable front end for querying and displaying the results.

Instructions:
1. The exercises are to be solved in an RDBMS environment like Oracle or DB2.
2. Suitable tuples have to be entered so that queries are executed correctly.
3. Front end may be created using either VB or VAJ or any other similar tool.
4. The student need not create the front end in the examination. The results of the queries may be displayed directly.
5. Relevant queries other than the ones listed along with the exercises may also be asked in the examination.
6. Questions must be asked based on lots.

SYSTEM SOFTWARE & OPERATING SYSTEMS LABORATORY

Subject Code: 10CSL58   **I.A. Marks : 25**
Hours/Week : 03   **Exam Hours: 03**
Total Hours : 42   **Exam Marks: 50**

PART - A
LEX and YACC Programs:
Design, develop, and execute the following programs using LEX:

1. a) Program to count the number of characters, words, spaces and lines in a given input file.

b) Program to count the numbers of comment lines in a given C program. Also eliminate them and copy the resulting program into separate file.

2. a) Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.
   b) Program to recognize whether a given sentence is simple or compound.

3. Program to recognize and count the number of identifiers in a given input file.

Design, develop, and execute the following programs using YACC:

4. a) Program to recognize a valid arithmetic expression that uses operators +, -, * and /.
   b) Program to recognize a valid variable, which starts with a letter, followed by any number of letters or digits.

5. a) Program to evaluate an arithmetic expression involving operators +, -, * and /.
   b) Program to recognize strings 'aaab', 'abbb', 'ab' and 'a' using the grammar ($a^n b^n$, n>= 0).

6. Program to recognize the grammar ($a^n b$, n>= 10).

PART B
UNIX Programming:
   Design, develop, and execute the following programs:

7. a) Non-recursive shell script that accepts any number of arguments and prints them in the Reverse order, (For example, if the script is named rargs, then executing rargs A B C should produce C B A on the standard output).
   b) 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.

8. a) Shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permissions

46

are identical, outputs the common permissions, otherwise outputs each file name followed by its permissions.

b) C program to create a file with 16 bytes of arbitrary data from the beginning and another 16 bytes of arbitrary data from an offset of 48. Display the file contents to demonstrate how the hole in file is handled.

9. a) Shell script that accepts file names specified as arguments and creates a shell script that contains this file as well as the code to recreate these files. Thus if the script generated by your script is executed, it would recreate the original files(This is same as the "bundle" script described by Brain W. Kernighan and Rob Pike in " The Unix Programming Environment", Prentice – Hall India).

b) 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 the wait( )) and prints its own process-id and the id of its child process and then exits.

Operating Systems:

10. Design, develop and execute a program in C / C++ to simulate the working of Shortest Remaining Time and Round-Robin Scheduling Algorithms. Experiment with different quantum sizes for the Round-Robin algorithm. In all cases, determine the average turn-around time. The input can be read from key board or from a file.

11. Using OpenMP, Design, develop and run a multi-threaded program to generate and print Fibonacci Series. One thread has to generate the numbers up to the specified limit and another thread has to print them. Ensure proper synchronization.

12. Design, develop and run a program to implement the Banker's Algorithm. Demonstrate its working with different data values.

Instructions:
In the examination, a combination of one LEX and one YACC problem has to be asked from Part A for a total of 30 marks and one programming exercise from Part B has to be asked for a total of 20 marks.