Code No	Subject		Theory			Practical		
		Lecture	Credit	University	Internal	Hours/ Week	Credit	Marks
		Hrs/ Week	Theory	Marks	Evaluation	L/T	Practical	
	Semester – 4							
NMCA401	Programming with Java	3	3	100	50	2	1	50
NMCA 402	Computer Graphics and Multimedia	3	3	100	50	2	1	50
NMCA 403	Software Engineering	3	3	100	50	2	1	50
NMCA 404	Compiler Design and Language Processor	3	3	100	50	2	1	50
NMCA 405	Personality and Soft Skill Development					6	2	150
NMCA 406	Elective I	3	4	100	50			
NMCA 407	Group Discussion/ Seminar					4	2	100
TOTAL		15	16	500	250	16	08	400
TOTAL Marks: 1200								
Total Credits: 24								

4th Semester detailed Syllabus

NMCA 401 PROGRAMMING WITH JAVA

Module 1 (10 Hours)
Features of Java, Data types, operators & expressions, control structures, arrays, Classes, objects & methods, constructors, garbage collection, access qualifiers, string handling – string operations, character extraction, string comparison, searching and modifying strings, String Buffer, packages and interfaces, Wrapper classes.

Module 2 (10 Hours)
Inheritance: single and multilevel inheritance, method overriding, abstract class, use of super and final keywords. Exception Handling: Exception types, uncaught exceptions, multiple catch clauses, nested try statements, built-in exceptions, creating your own exceptions. Multithreading: Java thread model, creating multiple threads, thread priorities, synchronization, interthread communication, suspending, resuming and stopping threads.

Module 3 (10 Hours)
Applets: Local & Remote Applets, Applet Architecture, Passing Parameters to Applets, Applet Graphics, Adapter Class. I/O Streams: Console I/O – reading console input, writing console output, Files I/O – Byte Streams, Character Streams, Collection Interfaces & Classes, Delegation Event Model

Module 4 (10 Hours)
AWT Classes: Window fundamentals, working with graphics, working with color & fonts. AWT controls, layout managers & working with menus, JFrames. Swing Classes, Java Beans, Servlet classes & Life Cycle.

Module 5 (6 Hours)
(as per choice of faculty)
Portion covered can be tested through Internal evaluation only not to be included in University examination)

Text Books:

1. Herbert Schildt, The Complete Reference Java 2, Fourth Edition, Tata McGraw Hill-2001
2. Liang Y.Daniel, Introduction to Java Programming (7th Edition), 2009, Pearson

1. Steven Holzner, Java 1.2, BPB-1998
2. E. Balaguruswami, Programming with Java - Second Edition, Tata McGraw Hill-1998.
3. Mughal K.A., Rasmussen R.W., A Programmer‟s Guide to Java Certification, Addison-Wesley, 2000 `

NMCA 402 COMPUTER GRAPHICS AND MULTIMEDIA

Module 1 (10 Hours)

An Introduction Graphics System : Computer Graphics and Its Types, Application of computer graphics, Graphics Systems : Video Display Devices, Raster Scan Systems, Random Scan Systems, Graphics Monitors and Work Stations, Input Devices, Hard Copy Devices, Graphics Software.

Module 2 (10 Hours)

Output Primitives and Attributes of Output Primitives : Output Primitive Points and Lines, Line Drawing Algorithms, Circle Generating Algorithms, Scan-Line Polygon Fill Algorithm, Inside-Outside tests, Boundary-Fill Algorithm, Flood Fill Algorithm, Cell Array, Character Generation, Attributes of Output Primitives : Line Attributes, Color and Grayscale Levels, Area fill Attributes, Character Attributes, Bundled Attributes, Anti-aliasing.

Module 3 (10 Hours)

Two-dimensional Geometric Transformations : Basic Transformations, Matrix Representation and Homogeneous Coordinates, Composite Transformations, Reflection and Shearing. Two-Dimension Viewing : The viewing Pipeline, Window to view port coordinate transformation, Clipping Operations, Point Clipping, Line Clipping, Polygon Clipping, Text Clipping, Exterior Clipping Three-Dimensional Concepts : Three Dimensional Display Methods, 3D Transformations, Parallel Proection and Perspective Projection.

Module 4 (10 Hours)

Multimedia : Introduction to Multimedia : Classification of Multimedia, Multimedia Software, Components of Multimedia – Audio : Analog to Digital conversion, sound card fundamentals, Audio play backing and recording Video, Text : Hypertext, Hyper media and Hyper Graphics, Graphics and Animation : Classification of Animation. Authoring Process and Tools. Case Study: graphics software MatLab, Use of MatLab in graphics application, Features of MatLab, Generalize application by using MatLab.

Module 5 (6 Hours)

(as per choice of faculty)

Portion covered can be tested through Internal evaluation only not to be included in University examination)

Text Books:

1. Donald **Hearn**& M. Pauline **Baker**, "*Computer Graphics with OpenGL*", Third Edition, 2004, Pearson Education, Inc. New Delhi.

2. Ze-Nian**Li** and Mark S. **Drew**, "*Fundamentals of Multimedia*", First Edition, 2004, PHI Learning Pvt. Ltd., New Delhi.

Reference Books:

1. Plastock : Theory & Problem of Computer Gaphics, Schaum Series.
2. Foley & Van Dam : Fundamentals of Interactive Computer Graphics, Addison-Wesley.
3. Newman : Principles of Interactive Computer Graphics, McGraw Hill.
4. Tosijasu, L.K. : Computer Graphics, Springer-Verleg.
5. S. Gokul : Multimedia Magic, BPB Publication.
6. Bufford : Multimedia Systems, Addison Wesley.
7. Jeffcoate : Multimedia in Practice, Prectice-Hall.
8. Any other book(s) covering the contents of the paper in more depth.

Note : Latest and additional good books may be suggested and added from time

NMCA 403 OBJECT ORIENTED SOFTWARE ENGINEERING

Module 1 (10 Hours)

Software Process Models:

Software Product, Software crisis, Handling complexity through Abstraction and Decomposition, Overview of software development activities, Process Models,
Classical waterfall model, iterative waterfall model, prototyping mode, evolutionary model, spiral model, RAD model, Agile models: Extreme Programming.

Module 2 (10 Hours)

Software Requirements Engineering:

Requirement Gathering and Analysis, Functional and Non-functional requirements, Software Requirement Specification (SRS), IEEE 830 guidelines, Decision tables and trees.

Software Project Management:
Responsibilities of a Software project manager, project planning, Metrics for project size estimation, Project estimation techniques, Empirical estimation techniques, COCOMO models, Scheduling, Organization &team structure, Staffing, Risk management, Software configuration management.

Module 3 (10 Hours)

Structured Analysis & Design:

Overview of design process: High-level and detailed design, Cohesion and coupling, Modularity and layering, Function–Oriented software design: Structured Analysis using DFD Structured Design using Structure Chart, Basic concepts of Object Oriented Analysis & Design. User interface design, Command language, menu and iconic interfaces.

Coding and Software Testing Techniques:

Coding, Code Review, documentation. Testing: - Unit testing, Black-box Testing, White-box testing, Cyclomatic complexity measure, coverage analysis, mutation testing, Debugging techniques, Integration testing, System testing, Regression testing.

Module 4 (10 Hours)

Software Reliability and Software Maintenance:

Basic concepts in software reliability, reliability measures, reliability growth modeling, Quality SEI CMM, Characteristics of software maintenance, software reverse engineering, software reengineering, software reuse.

Emerging Topics:

Client-Server Software Engineering, Service-oriented Architecture (SOA), Software as a Service (SaaS).

Module 5 (6 Hours)

(as per choice of faculty)

Portion covered can be tested through Internal evaluation only not to be included in University examination)

Text Books:

1. Fundamentals of Software Engineering, Rajib Mall, PHI, 2014.
2. Software Engineering, A Practitioner"s Approach, Roger S. Pressman, TMG Hill.

Reference Books:

1. Software Engineering, I. Somerville, 9th Ed. , Pearson Education.

NMCA 404 COMPILER DESIGN AND LANGUAGE PROCESSOR

Module 1 (10 Hours)

Introduction to Compilers: Compilers and translators, Phases of compiler design, cross compiler, Bootstrapping, Design of Lexical analyser, LEX programming.

Syntax Analysis: Specification of syntax of programming languages using CFG, Top-down parser, design of LL (1) parser, bottom up parsing technique, LR parsing algorithm, Design of SLR, LALR, CLR parsers.YACC programming.

Module 2 (10 Hours)

Syntax directed translation: Study of syntax directed definitions & syntax directed translation schemes, implementation of SDTS, intermediate notations: postfix, syntax tree, TAC, translation of expression, controls structures, declarations, procedure calls, Array reference.

Storage allocation & Error Handling: Run time storage administration, stack allocation, symbol table management, Error detection and recovery: lexical, syntactic, semantic.

Module 3(10 Hours)

Code optimization: Important code optimization techniques, loop optimization, control flow analysis, data flow analysis, Loop invariant computation, Induction variable removal, Elimination of Common sub expression.

Module 4 (10 Hours)

Code generation – Problems in code generation, Simple code generator, Register allocation and assignment, Code generation from DAG, Peephole optimization.

Module 5 (6 Hours)(As per choice of faculty)
(Portion covered can be tested through Internal evaluation only not to be included in University examination)

Text Books

1. Principles of Compiler Design by Alfred V. Aho., Jeffrey D. Ulman.
2. “Compilers: Principles, Techniques and Tools” Aho, Ravi Sethi, Ullman, Pearson Education, VIII Ed. 2002.

Reference Books

1. Lex and Yacc by Johan R. levine, Tonny Mason, et. al. O” Reilly and Assosiates.
2. “Compilers Design in C” Allen I. Holub, PHI eastern economy edition 2003.

NMCA 405 PERSONALITY AND SOFT SKILL DEVELOPMENT

(Practical)

LIST OF TASKS:

1. Listening comprehension – Achieving ability to comprehend material delivered at relatively fast speed; comprehending spoken material in Standard Indian English, British English, and American English; intelligent listening in situations such as interview in which one is a candidate.
2. Vocabulary building, Creativity, using Advertisements, Case Studies etc.
3. Personality Development: Decision-Making, Problem Solving, Goal Setting, Time Management & Positive Thinking
4. Cross-Cultural Communication: Role-Play/ Non-Verbal Communication.
5. Meetings- making meeting effective, chairing a meeting, decision-making, seeking opinions, interrupting and handling interruptions, clarifications, closure-Agenda, Minute writing.
6. Group Discussion – dynamics of group discussion, Lateral thinking, Brainstorming and Negotiation skills
7. Resume writing – CV – structural differences, structure and presentation, planning, defining the career objective
8. Interview Skills – formal & informal interviews, concept and process, pre-interview planning, opening strategies, answering strategies, interview through tele and video-conferencing
9. Writing Skills - Business Communication, Essays for competitive examinations.
10. Technical Report Writing/ Project Proposals – Types of formats and styles, subject matter – organization, clarity, coherence and style, planning, data-collection, tools, analysis.- Feasibility, Progress and Project Reports.
11. Very Similar Test of standard software companies like TCS, WIPRO, InfoSys, Googleetc
12. Brain teasing tests

REFERENCES:

1. Simon Sweeny, “English for Business Communication”, CUP, First South Asian Edition, 2010.
2. M. Ashraf Rizvi, “Effective Technical Communication”, Tata McGraw-Hill Publishing Company Ltd. 2005.
3. Dr A Ramakrishna Rao, Dr G Natanam& Prof SA Sankaranarayanan, “English Language Communication: A Reader cum Lab Manual”, Anuradha Publications, Chennai, 2006.
4. Dr. ShaliniVerma, “Body Language- Your Success Mantra”, S. Chand, 2006.
5. Andrea J. Rutherford, “Basic Communication Skills for Technology”, 2nd Edition, Pearson Education, 2007.
6. Sunita Mishra & C. Muralikrishna, “Communication Skills for Engineers”, Pearson Education, 2007.
7. Jolene Gear & Robert Gear, “Cambridge Preparation for the TOEFL Test”, 2010.
8. Meenakshi Raman&Sangeeta Sharma, “Technical Communication”, Oxford University Press, 2011.

NMCA 406A: EMBEDDED SYSTEM

Examples of Embedded systems and Typical hardware HardwareFundamentalsforSoftwareEngineerandAdvancedHardwareFundamentals Interrupts and Survey of software architectures. Introduction to RTOS and More Operating System Services Basic Design using RTOS. Embedded Software development tools and Debugging Techniques

Text Books:

1.AnEmbeddedSoftwarePrimer,DavidA.Simon,PearsonEducation,Inc.,1999
2.EmbeddedRealTimeSystemsprogramming,SriramVIyerandPankajGupta,TMH,2004

Reference Books:

1.EmbeddedSystemsDesign–AUnifiedHardware/SoftwareIntroduction,Frank V ahid/TonyGivargis,JohnWiley&Sons,Inc.,2002
2. EmbeddedSystems,Architecture,ProgrammingandDesign,RajKamal,TMH,2003

NMCA 406B: DATA MINING TECHNIQUES

DSS-Uses, definition, Operational Database.Introduction to DATA Warehousing.Data-Mart, Conceptof Data-Warehousing, Multi Dimensional Database Structures.Client/Server Computing Model &DataWarehousing.Parallel Processor & Cluster Systems.Distributed DBMS implementations.DATA Warehousing. Data Warehousing Components, Building a Data Warehouse, Warehouse Database,Mapping the Data Warehouse to a Multiprocessor Architecture, DBMS Schemas for Decision Support,Data Extraction, Cleanup & Transformation Tools, MetadataBusiness Analysis: Reporting & Query Tools & Applications. On Line Analytical Processing (OLAP).Patterns&Models.Statistics, Artificial Intelligence.Knowledge Discovery, Data

Mining, Introduction to Data-Mining, Techniques of Data Mining, DecisionTree, Neural Networks, Nearest Neighbor & Clustering. Genetic Algorithm, Rule Introduction, Selecting& using the right Techniques.Multimedia Data-Mining, Multimedia Databases, Mining Multimedia Data, Data-Mining and the worldWide Web, Web Data Mining, mining, Mining and Meta-Data, Data Visualization & overall Perspective,Data Visualization, Application of Data-MiningIntroduction to Data Mining and knowledge discovery in databases (KDD); Data miningprimitives, concepts, tasks and functionalities - concept learning, classification and prediction,association rule mining, clustering and anomaly detection; Data preparation - cleaning,transformation, reduction, discretization; Techniques, approaches and evaluation: Credibility,evaluation and comparison of data mining models; Association rule mining techniques - Apriori,Partition-based, FP-tree, Pincer-search; Supervised (inductive) learning - Decision table, rule,tree; Model tree, Baye‟s theorem, k-nearest neighbour, Regression, SVM; Unsupervised learning– Clustering Techniques - Partition, k-d tree, Hierarchical, Density, Grid, Advanced Databases:Text, Sequence, Image, etc.

References:

1. J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2007
2. I.H. Witten, E. Frank, Data mining: Practical Tools and Techniques with Java Implementations, Morgan Kaufmann 1999
3. P-N. Tan, V. Kumar and M. Steinbach: Introduction to Data Mining, Pearson, 2007
4. D. Hand, H. Mannila, P. Smyth, Principles of Data Mining, Indian reprint, PHI 2004

NMCA 406C: WIRELESS COMMUNICATION AND MOBILE COMPUTING

Mobile radio systems-, Paging systems, cordless telephone system, cellular telephone system,Cellular Concept: Frequency reuse, channel assignment, hand off, Interference and cell splitting,sectoring, Improving Coverage and capacity in Cellular systems. Propagation modeling:Outdoor/ Indoor Propagation models, Small scale Multipath propagation- Rayleigh fading,RiceanFading, Nakagami fading, Shadowing, lognormal shadowing fading model, outageprobability, coverage estimation under shadowing, and multipath fading. Wireless Networks802.11, frequency-hopping, encoding and modulation, MAC Layer Protocol Architecture

Multiple access with collision avoidance protocol, Virtual Carrier-Sensing, DCF Protocol, PCFOperation.

References:

1. Rappaport, Wireless communications: principal and practice , Pearson ed.
2. Matthew s. Gast, 802.11 wireless networks, O"reilly
3. Andrea Goldsmith ,Wireless communication , cambridge university press ed .
4. JochenSchiller , Mobile communications, phi/person edu., 2nd ed.,

NMCA 406D:ERP AND E-COMMERCE

UNIT-I

Introduction: What is E-Commerce, Forces behind E-Commerce Industry Framework, Brief history of E-Commerce,Inter Organizational E-Commerce Intra Organizational E-Commerce, and Consumerto Business Electronic Commerce, Architectural framework Network Infrastructure for E-CommerceNetwork Infrastructure for E-Commerce, Market forces behind I Way, Component of I wayAccess Equipment, Global Information Distribution Network, Broad band Telecommunication.

UNIT-II

Mobile Commerce: Introduction to Mobile Commerce, Mobile Computing Application, Wireless Application
Protocols, WAP Technology, Mobile Information Devices, Web Security
Introduction to Web security, Firewalls & Transaction Security, Client Server Network, Emerging Client Server Security Threats, firewalls & Network Security.

UNIT-III

Encryption World Wide Web & Security, Encryption, Transaction security, Secret Key Encryption, Public
Key Encryption, Virtual Private Network (VPM), Implementation Management Issues.

UNIT – IV

Electronic Payments Overview of Electronics payments, Digital Token based Electronics payment System, SmartCards, Credit Card I Debit Card based EPS, Emerging financial Instruments, Home Banking,Online Banking.

UNIT-V

Net Commerce EDA, EDI Application in Business, Legal requirement in E -Commerce, Introduction to supplyChain Management, CRM, issues in Customer Relationship Management.

Books:

1. Greenstein and Feinman, “E-Commerce”, TMH
2. Ravi Kalakota, Andrew Whinston, “Frontiers of Electronic Commerce”, Addision Wesley
3. Denieal Amor, “ The E-Business Revolution”, Addision Wesley
4. Diwan, Sharma, “E-Commerce” Excel
5. Bajaj & Nag, “E-Commerce: The Cutting Edge of Business”, TMH

NMCA 406E: PHP AND MY SQL

UNIT-1:

Introduction to PHP
Evaluation of PHP, Basic Syntax, Defining variable and constant, Php Data type, Operatorand Expression.

Decisions and loop:
Making Decisions, Doing Repetitive task with looping, Mixing Decisions and looping with Html.

UNIT-2:

Function: What is a function, Define a function, Call by value and Call by reference, Recursive function, StringCreating and accessing, String Searching & Replacing String, Formatting String, StringRelated Library function Array Anatomy of an Array, Creating index based and Associative array Accessing array, ElementLooping with Index based array, Looping with associative array using each () and foreach(),Some useful Library function.

UNIT-3:

Handling Html Form with Php
Capturing Form, Data Dealing with Multi-value filed, and Generating File uploaded form, redirecting a form after submission.

Working with file and Directories
Understanding file& directory, Opening and closing, a file, Coping, renaming and deleting afile, working with directories, Creating and deleting folder, File Uploading & Downloading.

UNIT-4:

Session and Cookie
Introduction to Session Control, Session Functionality What is a Cookie, Setting Cookieswith PHP. Using Cookies with Sessions, Deleting Cookies, Registering Session variables,Destroying the variables and Session.

UNIT-5:

Database Connectivity with MySql

Introduction to RDBMS, Connection with MySql Database, Performing basic database operation(DML) (Insert, Delete, Update, Select), Setting query parameter, Executing queryJoin(Cross joins, Inner joins, Outer Joins, Self joins.)

Exception Handling
Understanding Exception and error, Try, catch, throw. Error tracking and debugging.

References:

1. Learning PHP, MySQL, books by „ O‟ riley Press

NMCA 401 JAVA PROGRAMMING LAB

1. Programs to illustrate constructors.
2. Programs to illustrate Overloading & Overriding methods in Java.
3. Programs Illustrate the Implementation of Various forms of Inheritance. (Ex. Single, Hierarchical, Multilevel inheritance….)
4. Program which illustrates the implementation of multiple Inheritance using interfaces in Java.
5. Program to illustrate the implementation of abstract class.
6. Programs to illustrate Exception handling
7. Programs to create packages in Java.
8. Program to Create Multiple Threads in Java.
9. Program to Implement Producer/Consumer problem using synchronization.
10. Program to Write Applets to draw the various polygons.
11.Create and Manipulate Labels, Lists, Text Fields, Text Areas & Panels
12. Handling Mouse Events & Keyboard Events.
13. Using Layout Managers.
14.Create& Manipulate the Following Text Areas, Canvas, Scroll bars, Frames, Menus, Dialog Boxes.
15. Programs, which illustrate the manipulation of strings. a.
Ex. 1. Sorting an array of Strings.
1. Frequency count of words & Characters in a text.
16. Programs, which illustrate the use of Streams.
17. Java Program that reads on file name from the user and displays the contents offile.
18. Write an applet that displays a simple message.
19. Write an applet that computes the payment of a loan based on the amount of the loan, theinterest rate and the number of months. It takes one parameter from the browser: Monthlyrate; if true, the interest rate is per month; Other wise the interest rate is annual.
20. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttonsfor the digits and for the + - X % operations. Add a text field to display the result.
21. Write a Java program for handling mouse events.
22. Write a Java program for creating multiple threads
23. Write a Java program that correctly implements producer consumer problem using the conceptof inter thread communication.
24. Write a Java program that lets users create Pie charts. Design your own user interface (with
AWT)
25. Write a Java program that allows the user to draw lines, rectangles and ovals.
26. Write a Java program that illustrates how run time polymorphism is achieved.

TEXT BOOK

1. THE COMPLETE REFERENCE JAVA J2SE 5TH EDITION BY – HERBERT SCHILDT (TMH)

REFERENCE BOOKS

1. THE COMPLETE REFERENCE JAVA 2 (Fourth Edition) BY - PATRICK NAUGHTON & HERBETSCHILDT (TMH)

2. PROGRAMMING JAVA - DECKER&HIRSH FIELD VIKAS PUBLISKING (2001) (THOMSONLEARNING) (SECOND EDITON)
3. INTRODUCTION TO JAVA PROGRAMMING - Y.DANIEL LIANG PHI(2002)
4. OBJECT ORIENTED PROGRAMMING THROUGH JAVA 2 BY - THAMUS WU (Mc.Graw Hill)
5. JAVA 2 - DIETEL & DIETEL (PEARSON EDUCATION)
6. INTRODUCTION TO JAVA – BALA GURU SWAMY

7. INTRODUCTION TO PROGRAMMIND & OOD USING JAVA – JAINO NINE & FA HOSCH (JOHNWILEY)
8. STARTING OUT WITH JAVA – JONY GADDIS (DREAM TECH PRESS)

NMCA 402 COMPUTER GRAPHICS AND MULTIMEDIA LAB

1. Program using OpenGL library functions, to implement the basic primitives such as POINT, LINES, QUAD, TRIANGLES and POLYGON etc.
2. Program using OpenGL library functions, to implement the line chart as per user input. Input monthly data for period of one year.
3. Program to draw hard wired house by using basic primitives of OpenGL library functions.
4. Program by using OpenGL library functions, to implement the Digital Differential Analyser line drawing algorithm.
5. Program by using OpenGL library functions, to implement the Bresenham‟s Line drawing, Circle drawing, Mid-point Circle drawing and Mid-point Ellipsedrawing algorithms.
6. Program by using OpenGL library functions, to implement the Cohen-Sutherland Line clipping algorithm.
7. Program by using OpenGL library functions, to implement the Liang-Barsky Line clipping algorithm..
8. Program to demonstrate 2D and 3D transformations.
9. Window to Viewport Transformation
10. Splines Using OpenGL, 2D Animation

NMCA 403 SOFTWARE ENGINEERING LAB

Use of Rational Rose 2.0/Higher

Objectives:

1. To know about Phases in software development project, overview, need, coverage of topics
2. To assign the requirement engineering tasks
3. To perform the system analysis : Requirement analysis, SRS
4. To perform the function oriented diagram : DFD and Structured chart
5. To perform the user‟s view analysis : Use case diagram
6. To draw the structural view diagram : Class diagram, object diagram
7. To draw the behavioral view diagram : Sequence diagram, Collaboration diagram
8. To draw the behavioral view diagram : State-chart diagram, Activity diagram
9. To draw the implementation view diagram: Component diagram
10. To draw the environmental view diagram : Deployment diagram
11. To perform various testing using the testing tool unit testing, integration testing

EXPERIMENT-1

Aim: Phases in software development project, overview, need, coverage of topics
Tools/ Apparatus: None.
Procedure:
1) Open an appropriate software engineering guide and study the software development life cycle and related topics.
2) Study the need of the software engineering.
3) Study the coverage of topics such as life cycle models and their comparisons.

EXPERIMENT-2

Aim: To assign the requirement engineering tasks.
Tools/ Apparatus: None.
Procedure:
1) Identify the different requirement engineering tasks.
2) Assign these tasks to various students to set the ball rolling.
3) Ask the students to start working on the given tasks.

EXPERIMENT-3

Aim: To perform the system analysis : Requirement analysis, SRS
Tools/ Apparatus: None.
Procedure:
1) Assign the group of the students different tasks of system analysis.
2) Ask students to meet different users and start analysis the requirements.
3) Ask students to give presentations group-wise of their system requirements analysis.

EXPERIMENT-4

Aim: To perform the function oriented diagram : DFD and Structured chart
Tools/Apparatus: Rational Rose Software.
Procedure:
1) Identify various processes, data store, input, output etc. of the system and askstudents to analyse.
2) Use processes at various levels to draw the DFDs.
3) Identify various modules, input, output etc. of the system and ask students to analyse.
4) Use various modules to draw Structured charts.

EXPERIMENT-5

Aim: To perform the user‟s view analysis : Use case diagram
Tools/Apparatus: Rational Rose Software.
Procedure:
1) Identify various processes, use-cases, actors etc. of the system and ask students toanalyse.
2) Use processes at various levels to draw the use-case diagram.

EXPERIMENT-6

Aim: To draw the structural view diagram : Class diagram, object diagram
Tools/Apparatus: Rational Rose Software.
Procedure:
1) Identify various elements such as classes, member variables, member functions etc. of the class diagram
2) Draw the class diagram as per the norms.
3) Identify various elements such as various objects of the object diagram
4) Draw the object diagram as per the norms.

EXPERIMENT-7

Aim: To draw the behavioral view diagram : Sequence diagram, Collaboration diagram
Tools/Apparatus: Rational Rose Software.
Procedure:
1) Identify various elements such as controller class, objects, boundaries, messages etc.of the sequence diagram
2) Draw the sequence diagram as per the norms.
3) Identify various elements such as for the sequence diagram of the collaboration diagram
4) Draw the collaboration diagram as per the norms.

EXPERIMENT-8

Aim: To draw the behavioral view diagram : State-chart diagram, Activity diagram
Tools/Apparatus: Rational Rose Software.
Procedure:
1) Identify various elements states and their different transition of the state-chart diagram

2) Draw the state-chart diagram as per the norms.
3) Identify various elements such as different activity their boundaries etc. of the activity diagram
4) Draw the activity diagram as per the norms.

EXPERIMENT-9

Aim: To draw the implementation view diagram: Component diagram.
Tools/Apparatus: Rational Rose Software.
Procedure:
1) Identify various elements of the component diagram such as the various components like client, server, network elements etc.
2) Draw the component diagram as per the norms.

EXPERIMENT-10

Aim: To draw the implementation view diagram: deployment diagram
Tools/Apparatus: Rational Rose Software.
Procedure:
1) Identify various elements such as the hardware components of the deployment diagram
2) Draw the deployment diagram as per the norms.

EXPERIMENT-11

Aim: To perform various techniques for testing using the testing tool : unit testing, Integrationtesting
Tools/Apparatus: Win-runner.
Procedure:
1) Identify various modules of the system so that they can be tested stand alone.
2) Identify the groups of the module that can be tested together in integration.
3) Perform the testing of the modules as a unit and in integration by using the testing tool.

EXPERIMENT-12

Aim: To draw UML diagrams using Rational rose software. Tools/Apparatus: Rational rose software.
Procedure:
1) Identify various elements of the system to be drawn using the IDE.
2) Use the UML options of the rational rose to draw the diagrams from experiment 4 to 10.

EXPERIMENT-13

Aim: To draw UML diagrams using MS Visio software.
Tools/Apparatus: MS Visio software. Procedure:
1) Identify various elements of the system to be drawn using the IDE.
2) Use the UML options of the MS Visio software to draw the diagram from experiment 4 to 10.

Reference books:

1. Fundamentals of Software engineering,Rajib Mall.
3. Software design – From programming to architecture, Eric Braude
5. Object-oriented software engineering – A use case driven approach,Ivar Jacobson(Computer language productivity award winner)

NMCA 404 COMPILER DESIGN AND LANGUAGE PROCESSOR LAB

Practice of LEX and YACC in windows/Linux OS. Practice of writing of programs either in C/C++/JAVA for implementation.

List of Experiments:

1. Design a lexical analyzer for given language and the lexical analyzer should ignoreredundant spaces, tabs and new lines. It should also ignore comments. Although thesyntax specification states that identifiers can be arbitrarily long, you may restrict thelength to some reasonable value. Simulate the same in C/LEX language.
2. Write a program to identify whether a given line is a comment or not.
3. Write a program to recognize strings under 'a', 'a*b+', 'abb'.
4. Write a program to test whether a given identifier is valid or not.
5. Write a program to simulate lexical analyzer for validating operators.
6. Implement the lexical analyzer using JLex, flex or other lexical analyzer generating Tools.
7. Write a program for implementing the functionalities of predictive parser for the miniLanguageas specified in **Note 1**.
8. Write a program for constructing of LL (1) parsing
9. Write a program for constructing recursive descent parsing.
10. Write a program to implement LALR parsing.
11. Write a program to implement operator precedence parsing
12. Write a program to implement Program semantic rules to calculate theexpression that takes an expression with digits, + and * and computes the value.
13. Convert the BNF rules into Yacc form and write code to generate abstract syntax treefor the mini language
14. Write a program to generate machine code from abstract syntax tree generated by theparser. The instruction set specified in **Note 2** may be considered as the target code.

Note 1:

A simple language written in this language is

```
{int a[3],t1,t2;
T1=2;
A[0]=1;a[1]=2;a[t]=3;
T2=-( a[2]+t1*6)/(a[2]-t1);
If t2>5then
Print(t2)
Else{
Int t3;
T3=99;
T2=25;
Print(-t1+t2*t3);/*this is a comment on 2 lines*/
}endif
}
```

Comments(zero or more characters enclosed between the standard C/JAVA Style comment brackets/*…*/)can beinserted .The language has rudimentary support for1-dimenstional array, the declaration int a[3] declares an array ofthreeelements,referenced as a[0],a[1] and a[2].

Note:You should worry about the scoping of names.

Experiment with:

1. Write a program to compute FIRST for the following grammar?

E→ TE'
E' →+TE'/î
T →FT‟
T'→*FT'/î
F→ (E)/i

2. Write a program to compute FIRST for the following grammar?

S→iCtSS‟
S‟→eS/ î

3. Write a program to construct predictive parsing table for the following grammar?

S→iCtSS‟
S‟→eS/ î

Note 2:
Consider the following mini language, a simple procedural high –level language, only operating on integer data, with asyntax looking vaguely like a simple C crossed with Pascal. The syntax of the language is defined by the followinggrammar.

<program>::=<block>
<block>::={<variable definition><slist>}
|{<slist>}
<variabledefinition>::=int<vardeflist>
<vardec>::=<identifier>|<identifier>[<constant>]
<slist>::=<statement>|<statement>;<slist>
<statement>::=<assignment>|<ifstament>|<whilestatement>
|<block>|<printstament>|<empty>
<assignment>::=<identifier>=<expression>
|<identifier>[<expression>]=<expression>
<if statement>::=if<bexpression>then<slist>else<slist>endif
|if<bexpression>then<slisi>endif
<whilestatement>::=while<bexpreession>do<slisi>enddo
<printstatement>:;=print(<expression>)
<expression>::=<expression>::=<expression><addingop><term>|<term>|<addingop>
<term>

<bexprssion>::=<expression><relop><expression>
<relop>::=<|<=|==|>=|>|!=
<addingop>::=+|-
<term>::=<term><multop><factor>|<factor>
<Multop>::=*|/
<factor>::=<constant>|<identifier>|<identifier>[<expression>]
|(<expression>)
<constant>::=<digit>|<digit><constant>
<identifier>::=<identifier><letter or digit>|<letter>
<letter or digit>::=<letter>|<digit>
<letter>:;=a|b|c|d|e|f|g|h|I|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
<digit>::=0|1|2|3|4|5|^|7|8|9
<empty>::=has the obvious meaning

Experiment with:

1. Write a program to generate the code for the following three address code statements?
   A=B+C
   W=X-Y
2 Write a program to generate the code for the following three address code statements?
   W=(A+B)*C

NMCA 409 GROUP DISCUSSION/SEMINAR

Tasks:

Reading of newspapers, writing of articles, how to prepare seminars and reports, technical paper writing skills, GD on current topics, invited guest for spoken English, HR personnel from IT industries.

Note:10 things to succeed in Group Discussion

Read voraciously

Make a habit of reading voraciously on every subject. This will keep you ready for any topic for a discussion in GD Your knowledge is your most important weapon in a discussion.

Initiate the discussion

Most of us have a misconception that initiating the discussion would give you an advantage over others. It does give you an advantage but only if you know the subject well and have something relevant to start the discussion otherwise it is a disadvantage.
For e.g. when a group was given a subject “Is Capital punishment right?” some members of the group heard the word punishment and jumped at starting the discussion without understanding the meaning of Capital Punishment. The evaluators kept hearing for 2 minutes after which they intervened and asked the group if they knew the meaning of Capital Punishment. Not to say, the members who initiated were quite looking at each other‟s faces. That is when a quite member of the group got up and explained the meaning of the topic. From this incidence, you can easily tell who must have succeeded in the GD, the ones who initiated the discussion or the one who explained the topic and gave it a right direction.
They say, “Speaking just for the sake of speaking is noise”. So, don‟t create noise in the GD rather make some useful and resourceful contributions to get noticed in the discussion.

Speak politely and pleasantly

As you speak make sure that you do not speak at the top of your voice. You should be audible and clear. Remember that you are participating in a discussion which is different from a speech given out by the leaders in their rallies. Even if you disagree with the other‟s point of view, disagree politely. Use phrases like, I would like to disagree a bit here, I am sorry but I think I have a slightly different point of view here.

Be précised

Abstain from using irrelevant information and data from your talks during a GD Speak precisely so that others also get a chance to put across their point of view.

Acquire and apply knowledge

Stay attentive to the ideas put forward by other group members and keep writing the important points discussed during the GD As you get a chance to speak, put forward your views about the topic. You can also agree or disagree with other‟s ideas, based on your knowledge about the subject.

Agree with the right

Don‟ttake a stand on either extreme when the discussion begins. It might happen that you get convinced by other‟s argument and want to change your stand. Respect other‟s opinion as well and agree with what is right, even if you initially had a different opinion.

Speak confidently

Maintain your confidence as you speak. Establish eye contact with other members of the group and do not let your voice tremble.

Moderate

Try to moderate the discussion if any arguments arise. This is necessary to ensure that the group doesn‟t wander from the goal of the GD

Use positive body language

Your body language should not demonstrate dominance or low self-confidence. Show your interest in the discussion through your gestures like bending forward a bit, nodding your head.

Be a team player

Last but not the least; be a team player as this is a group activity. Be comfortable with the group members and vice versa.

Sample GD topics

- Reservation system should be stopped
- Donald Trump‟s presidency – Impact on India bad or good

- Divorce and remarriage should be encouraged
- Reservation for women would help the society
- Hindi movies are harming our society
- Live-in relationships should be encouraged
- India should be reorganized into smaller states
- IT boom and the growing pressure
- Smaller businesses and start-ups have more scope

- Developing countries need trade, not aid
- China is a threat to Indian IT industry
- Should agricultural subsidies be stopped?
- Multinational corporations: Are they devils in disguise?
- Business and Ethics do not go together
- Indian culture doesn't breed leaders
- India - really the NexGen superpower
- Fate of Apple after Steve Jobs
- FDI in Retail - Will really affect the farmers of India?
- EU Zone Crisis - reason for rising value of dollar
- US Debt Crisis - really has an impact on world market
- Should central government provide West Bengal a moratorium on loan repayments?
- Sanctions against Iran - right or wrong?
- FDI in Indian retail should be welcomed
- China market - a threat to Indian market
- Black money in tax heavens - declared national property
- Rising petrol prices - Govt. can control?
- Government should give up the control on CBI
- US war on Iraq-justified or not?
- Depreciation of Indian Rupee has only negative impact on the economy
- Nokia and Microsoft are a planned alliance or desperate move?
- RBI cannot control inflation with its temporary monetary policies
- Ditching the Kyoto Protocol - Is India's objection on EU justified?

Important:

Each student has to arrange summer training/internship in Industry or Educational Institute for 2 to three months duration or research work followed by depositing a project report and presentation in fifth semester. The internship shall be evaluated in fifth semester.