GOVERNMENT COLLEGE OF ENGINEERING, AURANGABAD

(An Autonomous Institute of Government of Maharashtra)

Department of Information Technology

Teaching and Evaluation Scheme

TE (Full-Time) in IT Engineering

SEMESTER-I

THEORY COURSES												
S. No.	**Course Code**	**Subject**	**Scheme of Teaching (Hrs/Week)**			**Total Credits**	**Scheme of Evaluation (Marks)**					
			L	**T**	**P**		**Theory**			**Term Work**	**Practical/Viva-voce**	**Total**
							Test	**TA**	**ESE**			
1	IT-341	Microprocessor and Interfacing	03	01		04	20	20	60			100
2	IT-342	Computer Algorithms	03	01		04	20	20	60			100
3	IT-343	Software Engineering and Testing	04			04	20	20	60			100
4	IT-344	Programming in Java	03			03	20	20	60			100
5	IT-345 IT-346	Elective – I Human Computer Interface Information Theory and Coding	04			04	20	20	60			100
LABORATORY COURSES												
1	IT-347	Lab: Microprocessor and Interfacing			02	01				25	25	50
2	IT-348	Lab: Computer Algorithm			02	01				25	25	50
3	IT-349	Lab: Software Engineering and Testing			02	01				25	25	50
4	IT-350	Lab.- Programming in Java			02	01				25	25	50
5	IT-351	Lab: Software Development Lab I(C Sharp)			02	01				25	25	50
			17	02	10	24	100	100	300	125	125	750

L-Lectures, T-Tutorials, P-Practical, TA-Teacher Assessment, ESE-End-Semester Examination

60 Contact hours for Four credits subject, 45 Contact hours for three credits subject, 30 Contact hours for two credits subject, 15 Contact hours for one credit subject

GOVERNMENT COLLEGE OF ENGINEERING, AURANGABAD

(An Autonomous Institute of Government of Maharashtra)

Department of Information Technology

Teaching and Evaluation Scheme

TE (Full-Time) in IT Engineering

SEMESTER-II

THEORY COURSES												
S. No.	**Course Code**	**Subject**	**Scheme of Teaching (Hrs/Week)**			**Total Credits**	**Scheme of Evaluation (Marks)**					
			L	**T**	**P**		**Theory**			**Term Work**	**Practical /Viva-voce**	**Total**
							Test	**TA**	**ESE**			
1	IT-352	Theory of Computation	03	01		04	20	20	60			100
2	IT-353	Computer Networks	03	01		04	20	20	60			100
3	IT-354	Advance Database Management System	04			04	20	20	60			100
4	IT-355	Operating System	03			03	20	20	60			100
5	IT-366 IT-357	Elective –II Advanced Java Object Oriented Modeling and Design	04			04	20	20	60			100
LABORATORY COURSES												
1	IT-358	Lab: Computer Networks			02	01				25	25	50
2	IT-359	Lab: Advance Database Management System			02	01				25	25	50
3	IT-360	Lab: Software Development Lab-II(ASP.NET)			02	01				25	25	50
4	IT-361	Lab: Operating System			02	01				25	25	50
5	IT-362 IT-363	Elective-II Lab: Advanced Java Lab: Object Oriented Modeling and Design			02	01				25	25	50
			17	02	10	24				125	125	750

L-Lectures, T-Tutorials, P-Practical, TA-Teacher Assessment, ESE-End-Semester Examination

60 Contact hours for Four credits subject, 45 Contact hours for three credits subject, 30 Contact hours for two credits subject, 15 Contact hours for one credit subject

IT 341: Microprocessor and Interfacing

Teaching Scheme		**Evaluation Scheme**	
Lectures	03Hrs/Week	Test	20 Marks
Tutorials	01Hrs/Week	Teacher Assessment	20 Marks
Total Credits	04	End-Semester Examination	60 Marks

Contact Hours for this subject is 60

Course Educational Objectives:

1. To learn the architecture and assembly language programming 0f 8086
2. To learn peripherals and their interfacing with 8086 Microprocessor.
3. To study the DOS Internals.
4. To Study NDP and Design of Microprocessor based System

Course Outcomes Expected:

1. Student will able to design small model by using assembly level language.
2. Understand the concept of DMA controller and other interfaces for implementing real applications.

UNIT-1 Introduction to 16 bit microprocessor, Architecture and Pin diagram of 8086, Programmers model of 8086 (Registers), Segmentation, logical to physical address translation, even and odd memory banks, Read write cycle timing diagrams, Address mapping and decoding, I/O: memory mapped I/O & I/O Mapped I/O.

UNIT-2 Addressing modes, Instruction set of 8086 in detail, Instruction Formats, Stacks, Assembly Language Programming, Assembler, Linker, Debugger (Turbo debugger), Directives, Procedures (Near & Far), Macros, Loop constructs, 8086 Programming examples.

UNIT-3 8086 Interrupt Structure, Interrupt Vector Table (IVT), ISR, Hardware and software Interrupts Internals of DOS, DOS loading, DOS memory map, Internal and external commands of DOS, BIOS & DOS Interrupts. Concepts of PSP, .EXE & .COM files, Concepts of TSR, 8259 (Programmable Interrupt Controller): Features, Block Diagram, Control & status registers, Interfacing & Programming.

UNIT-4 Study of Peripheral chips: 8255 (Programmable Peripheral Interface), Serial Communication- Synchronous & Asynchronous, 8251(USART): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming (8255 and 8251)

Concept of ADC -Successive Approximation & Interfacing, Concept of DAC R-2R (ladder) & Interfacing, Introduction to Sensors & Transducers, Keyboard Display & Centronics Printer Parallel Interfacing using 8255.

UNIT-5 8279 Keyboard and Display Controller, 8253 (Programmable Interval Timer): Features, Block Diagram, Control & status registers, Operating modes, Interfacing & Programming, Concept of DMA, 8237 DMA Controller: Features, Block Diagram.

Minimum & Maximum mode of 8086, Support chips 8282,8284,8286,8288 8087(NDP) - Features, Block Diagram, Control & status registers, typical Instruction Set & Programming, Detail Design of 8086 based minimum system with EPROM, SRAM & Peripherals such as 8255,8253,8251,8279 with keyboard & seven segments Display.

Text Books:

1. Douglas Hall, “Microprocessors & Interfacing”, McGraw Hill, Revised 2nd Edition, 2006 ISBN 0-07-100462-9
2. John Uffenbeck,” The 8086/88 Family: Design, Programming & Interfacing”, PHI,
3. Ray, K.Bhurchandi, ”Advanced Microprocessors and peripherals: Arch, Programming & Interfacing”,Tata McGraw Hill,2004 ISBN 0-07-463841-6

References Books:

1. Liu, Gibson, “Microcomputer Systems: The 8086/88 Family”, 2nd Edition, PHI,2005
2. Kenneth Ayala, “The 8086 Microprocessor: Programming & Interfacing the PC”, Cengage Learning, Indian Edition, 2008
3. Ray Dunkon, “Advanced MSDOS Programming”, 2nd Edition, BPB Publication.
4. Kip Irvine, ”Assembly language for IBM PC”, PHI, 2nd Edition, 1993
5. Peter Abel, ”Assembly language programming”, Pearson Edu,5th Edition,2002
6. Intel Microprocessor and peripheral Handbook: Volume 1
7. Yashwant Kanetkar, “Writing TSR through C”, BPB Publication, 1995, ISBN 81-7029-520-

IT 342: Computer Algorithms

Teaching Scheme		**Evaluation Scheme**	
Lectures:	3 Hrs/Week	Test	20 Marks
Total Credits:	4	Teacher Assessment	20 Marks
Tutorial:	1 Hrs/Week	End-Semester Examination	60 Marks

Course Outcomes:

1. Explain the asymptotic notations. Describe Big O, omega and theta notations for analysis.
2. Practice finding the time complexity of different sorting techniques.
3. Show hand run of algorithms on search techniques and practically implement them.
4. Evaluate recurrence relations for logarithmic, quadratic, cubic and exponential time complexity.

Contact Hours for this subject is 60

UNIT-1 Introduction

Definition of an Algorithm, Algorithm Specification, Performance Analysis: Space and Time Complexity, Asymptotic Notation, Practical Complexities, A Brief Review of Elementary Data Structures: Stacks, Queues, Trees, Priority Queues, Sets and Disjoint Set Union, Graphs.

UNIT-2 Divide And Conquer

General Method of Divide And Conquer, Binary Search, Finding The Maximum and Minimum, Merge Sort, Quick Sort, Selection, Strassen's Matrix Multiplication.

UNIT-3 The Greedy Method

General Method, Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum Cost Spanning Trees, Optimal Storage on Tape, Optimal Merge Patterns, Single Sources Shortest Paths.

UNIT-4 Dynamic Programming

The General Method of Dynamic Programming, Multistage Graphs, All Pairs Shortest Paths, 0/1 Knapsack, the Travelling Salesperson Problem.

Basic Search And Traversal Techniques

Techniques For Binary Trees, Techniques For Graphs, Connected Components And Spanning Trees, Biconnected Components And DFS

UNIT-5 Backtracking And Branch And Bound

The General Method Of Backtracking, The 8- Queens Problem, Sum Of Subsets, Graph Coloring, Hamiltonian Cycles, Knapsack problem, The Method Of Branch And Bound, 0/1 Knapsack Problem, Travelling Sales Person Problem Using Branch And Bound.

TEXT AND REFERENCE BOOKS

1. Ellis Horowitz, Sarataj Sahni, S.Rajasekaran, "*Fundamentals of Computer Algorithms*", Galgotia Publications
2. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman , "*The Design and Analysis of Computer Algorithms*",
3. Addison WesleyCoreman, Leiserson, Rivest, Stein, *"Introduction to Algorithms",* PHI

IT 343 - SOFTWARE ENGINEERING AND QUALITY ASSURANCE

Teaching Scheme		**Evaluation Scheme**	
Lectures	04Hrs/Week	Test	20 Marks
		Teacher Assessment	20 Marks
Total Credits	04	End-Semester Examination	60 Marks

Contact Hours for this subject is 60

Course Educational Objectives:

1. To understand fundamental concepts of software engineering principles.
2. to design and implement the software solutions and methodologies, good test cases
3. To understand the role and contents of testing activities in different life cycle phases.
4. To train the students on basic principles of Software Engineering used in Industry.

Course Outcomes:

1. Able to design and apply software engineering principles, tools and techniques to develop, maintain and evaluate software solutions
2. ability to work as an effective member or leader of software engineering teams and meet ethical standards with legal responsibilities
3. ability to manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals

UNIT-1 **Introduction, Life Cycle Models:** Software engineering, Role of Management in software Development. SDLC Models, Selection of Life Cycle Model, Requirement Engineering, Requirements Elicitation, Requirement analysis, Requirements documentation,

Software Project Planning Size estimation, Cost estimation, Models, The constructive Cost mode COCOMO II, Software Risk management.

UNIT-2 **Software Design**

Design, Modularity, Strategy of design, Function oriented design, IEEE Recommended practice for Software Design Description , Object Oriented Design, Unified Modeling Language : Basic Notations, Class diagram, State diagram activity diagrams, Use-case diagrams sequence diagrams.

Software Metrics and Reliability Software metrics, data structures metrics, Information flow metrics, Metrics analysis, Software quality, software reliability Models, capability maturity Model, ISO 9000

UNIT-3 **Introduction to Basic of software testing & Terminology:** Quality Concepts, Quality Assurance, Quality Control, Needs of testing, Objective of testing, Software Development & Software Testing Life Cycle, Testing Standards:-IEEE, CMM,ANSI.

Levels of Testing: Verification and Validation Model, Techniques of Verification:-Peer Review, Walkthrough, Inspection. Functional testing, Structural testing, Unit testing, Integration testing, System testing, Installation Testing, Usability Testing, Regression testing, Performance testing, Load Testing**,** Stress Testing**.** Security testing**,** Volume testing Acceptance testing ,Alpha testing, Beta testing, Gamma testing, Object –oriented testing, Web testing, GUI testing

UNIT-4 **Testing methods and Testing tools:** Black Box methods**:-**Equivalence partitioning**,** Boundary-value analysis**,** Error guessing. White Box methods**:-**Statement coverage**,** Decision coverage**,** Condition coverage.

Testing Tools:-Win Runner, Load Runner.

UNIT-5 **Test Planning & Documentation:** Testing Strategy**:-**type of project, type of software**.** Test Plans, Test Case, Test Data, Risk Analysis.

Defect Management and Test Reporting: Defect analysis, Defect Reporting, Tracking Workflow, Test reporting, Defect rates and schedules Software maintenance, Maintenance process, maintenance models, estimation of maintenance costs, reverse engineering, software re-engineering, Configuration management, documentation.

Text books:

1. Pressman R.S., “ *Software Engineering*”, McGraw-Hill Publication
2. K.K.Aggarwal , Yogesh Singh , “*Software Engineeri*ng” , New Age International Publishers
3. Ron Patton, “*Software testing*”, Pearson Publications

Reference Books:

1. Ian Sommerville , “*Software Engineering*” , Pearson Education
2. Boris Bezier, “*Software testing techniques*”, Dreamtech Publications

3. Meilir Page-Jones, “ *Fundamentals of Object Oriented Design in UML*” Pearson Education
4. Rex Black , “*Software testing*”, Wrox Publications
5. Dr.K.V.K.K. Prasad, “*Software testing tools*”, Dreamtech Publications

Reference Website:
www.onestoptesting.com
www.wikipedia.org

IT 344: Programming in Java

Teaching Scheme		**Evaluation Scheme**	
Lectures	03Hrs/Week	Test	20 Marks
Tutorials	00Hrs/Week	Teacher Assessment	20 Marks
Total Credits	03	End-Semester Examination	60 Marks

Contact Hours for this subject is 45

Educational Objectives:

1. To enable the students to understand the core principles of the Java Language.
2. To understand the concept of Interfaces, Packages.
3. To understand the concept of Exception Handling and Multithreading.
4. Introduce the students to use visual tools to produce well designed, effective applications using applets and AWT/Swing.
5. Introduce the students to database connectivity.

Course Outcome Expected:

1. Develop Object oriented programs using java programming language.
2. Develop programs using Inheritance, Interfaces, Multithreading and Exception handling.
3. Develop user interface application using AWT/Swing.
4. Develop JDBC applications to access/query the database.

UNIT-1 Introduction to Java:

Why Java is important to the Internet? Java's Magic: The Byte Code, Java Buzzwords, Data types, Basic syntax of Java. Classes & Objects – Constructors, Access Modifiers, Instance Methods, this & static keywords. Overloading Methods, Overloading Constructors, Using objects as Parameters, A closer look at argument passing, Returning objects, Command Line Arguments.

Objectives: Create object-oriented programs using Java programming language

UNIT-2 Inheritance and Exception handling:

Inheritance :Basics, Using Super, Types of inheritance, Constructors in Derived Classes, Method Overriding, Abstract Classes & methods, Final Classes & Final Methods, Packages, Importing Packages and Interfaces. Exception Handling: Fundamentals of Exception handling, Exception Types, Uncaught Exceptions, Using Try and Catch, Multiple Catch Clauses, Throw, throws, finally, Built-in Exceptions ,user defined Exceptions

Objectives:

Create Java classes by extending existing Java classes

Implement Interfaces in Java application

Apply Exception Handling mechanism in Java application

UNIT-3 I/O Package and Multithreading:

I/O Package:File class, Various Methods of File class, Files and Directories, Overview of Streams, File Stream classes, Print Writer Class, Byte Stream classes Multithreading: Java Thread Model, The Main thread, Creating a Thread, Creating Multiple Threads, Using Alive ( ) and Join ( ), Thread Priorities, Synchronization, Interthread Communication,Suspending, Resuming and Stopping Threads.

Objectives: Create Multithreaded application in Java

UNIT-4 Java AWT/Swing:

Graphical Programming - Applet, Event Handling. Abstract Windows Toolkit- Components and Graphics, Containers, Frames and Panels Layout Managers- Border layout, Flow layout, Grid layout, Card layout AWT all components, Event delegation Model - Event source and handler, Event categories, Listeners, Interfaces.

Objevctives

Develop user interface applications using AWT/Swing.

Develop event-driven GUI interfaces using Java technology GUI components such as panels, buttons, labels, text fields and text areas

Apply layout manager to develop event-handling in Java

UNIT-5 **Java database connectivity:**

JDBC and Database Programming: Introduction to JDBC, JDBC Drivers, creating DSN, The java.sql package, PreparedStatement class, CallableStatement object, Scrollable Resultset, Updatable Resultset

Objevctives

Develop JDBC applications to access and query a database

TEXT AND REFERENCE BOOKS

Text Books:

1. Herbert Schildt, *"The Complete Reference Java2",* 5th Edition, TMH Publications.
2. Deitel & Deitel, *"How To Program JAVA",* Pearson Education
3. E Balguruswamy, *"Programming with Java - A Primer"*

Reference Books:

1. Cay S. Horstmann, Gary Cornell*, "Core Java Volume II"* - Pearson Education.
2. Steven Holzner, "*Java 2 Black Book*", Dreamtech Pub.

Reference websites:

www.java.sun.com/docs/books/tutorial

IT 345: Human Computer Interface

Teaching Scheme		**Evaluation Scheme**	
Lectures	4 Hrs/Week	Test	20 Marks
		Teacher Assessment	20 Marks
Total Credits	04	End-Semester Examination	60 Marks

Contact Hours for this subject is 60

Course Outcomes Expected:

At the end of the course the student will

1. realize the importance of user-centered design, design prototyping
2. explain the Theories and Principles for human and computer interface
3. understand design processes an tools
4. design interactive GUI objects

UNIT-1 Overview of HCI, Theories and Principles

Introduction, Goals of System Engineering, Goals of User-Interface Design Usability of Interactive systems, Motivations for Human Factors in Design, Guidelines, Principles, Theories, Conceptual, Semantic, Syntactic and Lexical Model, GOMS and the Keystroke-level Model, Object-Action Interface Mode

UNIT-2 Managing Design Processes and Tools and Testing

Three pillars of Design, Development Methodologies, Ethnographic Observation, Participatory Design, Scenario Development, Expert Reviews, Usability Testing and Laboratories, Acceptance Tests, Evaluation during active use, Specification Methods, Interface Building Tools, and Evaluation Tools

UNIT-3 Design Principles for Designing GUI Objects

Direct manipulation (examples, explanations), Visual Thinking and Icons, 3D Interfaces, Virtual Reality, Introduction to Menu Selection, Form Fill-in, and Dialog Boxes, Task Related Organizations, Fast Movement through Menus, Item Presentation Sequence, Response Time and Display Rate, Data Entry with Menus, Menu Layout, Command-Organizational Strategies, Naming and Abbreviations, Command Menus, Natural Language in Computing

UNIT-4 Interaction Styles

Introduction to Interaction Devices, Keyboards and Function Keys, Pointing devices, Speech and Auditory Interfaces, Speech Recognition, Image and video displays, Printers, Response time and display rate with respect to display, Goals of Collaboration, Asynchronous and Synchronous Interfaces, Face-to-Face Interfaces

UNIT-5 Presentation Design Issues and Information Search and visualization

Error Messages, Display Design, Individual-Window Design, Multiple Window Design, Co-ordination by Tightly-coupled Windows, Color, Printed Versus Online Manuals, Preparation of Online facilities, Online Tutorials, Online Communities for User Assistance

Introduction, Search in Textual Documents and Database Querying, Multimedia Document Searches, Advanced Filtering and Search Interfaces, Information Visualization, OAI Model for Website Design

TEXT AND REFERENCE BOOKS

- Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, "Designing the User Interface: Strategies for Effective Human-Computer Interaction", 5th Edition, Pearson Education
- Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale, "Human-Computer Interaction", 3rd Edition , Prentice Hall
- Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon (1994): Human-computer interaction. Addison-Wesley, ISBN 0201627698.
- Debbie Stone, Caroline Jarrett, Mark Woodroffe, Shailey Minocha (2005): User Interface Design and Evaluation. Morgan Kaufmann, ISBN 978-0120884360

IT346: Information theory and coding

Teaching Scheme		**Evaluation Scheme**	
Lectures	04Hrs/Week	Test	20 Marks
Tutorials	__Hrs/Week	Teacher Assessment	20 Marks
Total Credits	04	End-Semester Examination	60 Marks

Contact Hours for this subject is 60

Course Outcomes:

1) student will be able to define channel capacities and properties using Shannon's Theorems
2) student will be able construct efficient codes for data on imperfect communication channels
3) student will be able generalize the discrete concepts to continuous signals on continuous channels
4) Student will be able to describe the information resolution, compression, and efficient coding properties of wavelets.

UNIT-1 Information Theory: Introduction, Measure of information, Average information content of symbols in long independent sequences, Mark-off statistical model for information source, Entropy and information rate of mark-off source. Source Coding Uncertainty and information, average mutual information and entropy, information measures for continuous random
Variables, source coding theorem, Huffman codes.

UNIT-2 Source Coding: Encoding of the source output, Shannon's encoding algorithm,Communication Channels, Discrete communication channels, Continuous channels, Channel Capacity And Coding, Channel models, information capacity theorem, The Shannon limit. Linear And Block Codes For Error Correction

UNIT-3 Fundamental Limits on Performance: memory less Channels, Mutual information, Cyclic Codes Polynomials, division algorithm for polynomials, a method for generating cyclic codes, matrix description of cyclic codes,BCH Codes, Differential entropy and mutual information for continuous ensembles, Channel capacity Theorem Primitive elements, minimal polynomials, generator polynomials in terms of minimal polynomials, examples of BCH codes,Convolutional Codes

UNIT-4 Introduction to Error Control Coding: Types of errors, examples, Types of codes Linear Block Codes: Matrix description, Error detection and correction, Standard arrays and table look up for decoding,Binary Cycle Codes, Syndrome calculation. Tree codes, trellis codes, distance notions for convolutional codes, decoding of convolutional codes, distance and performance bounds for convolutional codes, examples of convolutional codes, Turbo codes, Turbo decoding.

UNIT-5 RS codes, Golay codes, Shortened cyclic codes, Burst error correcting codes. Burst and Random Error correcting codes.Convolution Codes, Time domain approach. Transform domain approach.

TEXT AND REFERENCE BOOK

1. Information theory, coding and cryptography - Ranjan Bose; TMH.
2. Information and Coding - N Abramson; McGraw Hill.
3. Introduction to Information Theory - M Mansurpur; McGraw Hill.
4. Information Theory - R B Ash; Prentice Hall.
5. Digital and analog communication systems, K. Sam Shanmugam, John Wiley, 1996.
6. Digital communication, Simon Haykin, John Wiley, 2003
7. ITC and Cryptography, Ranjan Bose, TMH, II edition, 2007
8. Digital Communications - Glover and Grant; Pearson Ed. 2nd Ed 2008

IT 347 : Lab-Microprocessor and Interfacing

Teaching Scheme		**Evaluation Scheme**	
Practical	2 Hrs/Week	Term Work	50 Marks
Credits	1	Practical/Viva-voce	--

Contact Hours for this subject is 15

Course Outcomes Expected

1. Demonstrate Arithmetic operators by using MASM
2. Understand the use of BCD converter
3. Implement the use of DOS/BOS programming
4. Understand the use of Interfacing by using various ICs

The term work shall consist of following practicals

I. Microprocessor 8086 :
1. Introduction to MASM/TASM.
2. Arithmetic operation – Multi byte Addition and Subtraction, Multiplication and Division – Signed and unsigned Arithmetic operation, ASCII – arithmetic operation.
3. Logic operations – Shift and rotate – Converting packed BCD to unpacked BCD, BCD to ASCII conversion.
4. By using string operation and Instruction prefix: Move Block, Reverse string, Sorting, Inserting, Deleting, Length of the string, String comparison.
5. DOS/BIOS programming: Reading keyboard (Buffered with and without echo) – Display characters, Strings.

II. Interfacing :
1. 8259 – Interrupt Controller : Generate an interrupt using 8259 timer.
2. 8279 – Keyboard Display : Write a small program to display a string of characters.
3. 8255 – PPI : Write ALP to generate sinusoidal wave using PPI.
4. 8251 – USART : Write a program in ALP to establish Communication between two processors.

TERM Work:

The term work consists of at least 10 experiments/ assignments based on the syllabus of the subject.
Assessment of term work should be done based on the points mentioned below:

- Continuous lab assessment
- Actually performing practices in the laboratory
- Oral Evaluation conducted (internally) at the time of submission

IT 348: Computer algorithms

Teaching Scheme		**Evaluation Scheme**	
Practical	2 Hrs/Week	Term Work	25 Marks
Credits	2	Practical/Viva-voce	25 Marks

Course Outcomes: Students will be able to

1. Design and implement appropriate data structures for solving computing problems.
2. Demonstrate algorithms using divide and conquer approach, greedy method.
3. Employ dynamic programming techniques.
4. Illustrate backtracking method to solve problems.

1 Implement recursive and iterative algorithms for specific problems.

2 Implementation of sorting methods using recursion.

3 Write a program for finding maximum and minimum using divide and conquer approach

4 Implement merge sort using divide and conquer approach.

5 Implement the Greedy Knapsack problem.

6 Write a program for finding minimal spanning Trees using Prim's/ Kruscal's Greedy approach.

7 Write a program for finding shortest path using multistage graph problem.

8 Implement the all pairs shortest path problem using dynamic programming approach.

9 Implement the different techniques for tree traversals

10 Implement the graph traversal techniques

11 Illustrate 8-Queens problem using general backtracking method and recursive backtracking method.

12 Write a program for travelling salesperson problem algorithm using
(a) Dynamic programming approach, and (b) Backtracking

*Vague statements such as "*Term work shall consist of assignments based on syllabus"* shall be avoided. This format can be changed if necessary (as in case of seminar, dissertation, etc.)

IT 349 - Lab: Software Engineering and Testing

Teaching Scheme		**Evaluation Scheme**	
Practical	02Hrs/Week	Term Work	25Marks
Credits	01	Practical/Viva-voce	50Marks

Contact Hours for this subject is 15

Course Outcomes:

1. Hands on software engineering principles, tools and techniques to develop, maintain and evaluate software systems
2. ability to design and develop efficient, reliable, robust and cost-effective software solutions
3. ability to work as an effective member or leader of software engineering teams and meet ethical standards with legal responsibilities

Computer Usage / Lab Tool

Win Runner/ IBM Rational Suite/Data Modeling tools/ Rationale's Software Engineering tools

Suggestive List of Experiments:

1 Program Analysis and project planning through the study of problem
Identify project scope, objectives and Infrastructure.
2 Software requirement analysis , describe the individual phases/ Module of the project,
Identify Deliverables.
3 Software design – use work product – data dictionary diagram, activity diagram, build and test class diagram, sequence diagram, DFD diagram, ER diagram
4 Software development and debugging using and tool (front end and back end)
5 Software verification and validation procedure
6 Study of Testing tools
7 Introduction to Win runner / rational rose/load runner.
8 Recording test in analog and context sensitive mode
9 Synchronizing test
10 Checking GUI Objects and bitmap objects
11 Programming test with TSL
12 Creating data driven test
13 Maintaining test script
14 Project (Creating test report)

*Vague statements such as "*Term work shall consist of assignments based on syllabus"* shall be avoided. This format can be changed if necessary (as in case of seminar, dissertation, etc.)

IT 350: Lab Programming in Java

Teaching Scheme		**Evaluation Scheme**	
Practical	02Hrs/Week	Term Work	25Marks
Credits	01	Practical/Viva-voce	50Marks

Contact Hours for this subject is 15

Course Outcomes Expected:

1 Demonstrate Java syntax and semantics.
2 Demonstrate Inheritence and Multithreading.
3 Demonstrate Exception handling.
4 Demonstrate Swing/Applet programing.

Suggestive List of experiments:

1 Install the JDK,set the variables and write a program to print “Hello” word in Java.

2 Program to find factorial of number using command line argument.

3 Program to find area of circle, square and rectangle using method overloading.

4 Program to design class Account using inheritance and static that show all functions of bank(Withdraw ,Deposit).

5 Program to design a class using abstract method and classes.

6 Program which use try & catch for Exception handling.

7 Program to write a java Applet that create some text fields and text areas to demonstrate features of each.

8 Program to handle user defined exception using throw keyword.

9 Program to implement interthread communication.

10 Program to demonstrate system clock.

11 Program to implement Flow layout and Border layout.

12 Mini Project on database connectivity.

IT 351: Lab Software Development Lab I(C Sharp)

Teaching Scheme		**Evaluation Scheme**	
Practical	02Hrs/Week	Term Work	25Marks
Credits	01	Practical/Viva-voce	50Marks

Contact Hours for this subject is 15

Course Outcomes:

1. Implement object oriented principles
2. Resolve run-time errors using Exception Handling techniques
3. Interpret Events handling techniques for interaction of the user with GUI
4. Develop Advance programs using C# which is required in IT industry**.**
5. Illustrate real life usage of different Microsoft technologies**.**

Suggestive List of experiments:

1	Demonstrate use of Constructors, Destructors and Garbage Collector
2	Demonstrate use of Interfaces and Abstract classes
3	Demonstrate use of Namespaces, Assemblies, Reflection
4	Demonstrate Exception Handling in C#
5	Demonstrate use of rich text control
6	Demonstrate use ListBox and ComboBox control
7	Demonstrate use of ListView control
8	Demonstrate use of TreeView control to create explorer
9	Demonstrate use of LinkLabel to establish connection between two forms
10	Demonstration of Microsoft Technologies

IT 352: Theory of Computation

Teaching Scheme		**Evaluation Scheme**	
Lectures:	3 Hrs/Week	Test	20 Marks
Total Credits:	4	Teacher Assessment	20 Marks
Tutorial :	1 Hrs/Week	End-Semester Examination	60 Marks

Prerequisites: Data Structures

Course Outcomes: This course aims to make students understand basic concepts in the theory of computation.
After studying this course students will be able to:
1 Design finite automata and its equivalent regular expressions.
2 Convert DFA's to NFAs and regular expression.
3 Show certain languages are not regular.
4 Formulate pushdown automata and its equivalent context free grammars.
5 Use pumping lemmas to prove the type of language.
6 Construct Turing machines.

Contact Hours for this subject is 60

UNIT-1 **Mathematical Preliminaries and Finite Automata:** Sets, Relations and Functions, Principle of Induction, Introduction to Finite Automata, Structural Representations, Automata and Complexity, Central Concepts to Automata Theory, Deterministic finite Automata (DFA), Nondeterministic finite Automata (NFA), Equivalence of DFA and NFA, FA with epsilon transition, Applications of FA, Moore and Mealy machines.

UNIT-2 **Regular Expressions and Languages:** Regular Expressions, Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Proving languages not to be regular, Closure Properties of Regular Expressions, Decision Properties of Regular Expressions, Equivalence and minimization of Automata.

UNIT-3 **Context Free Grammar:** Definition, Derivations using grammar, Language of a grammar, Parse Trees, Application of CFG, Ambiguity in Grammars and Languages, Normal Forms for Context Free Grammars, Pumping Lemma for CFL, Closure and Decision Properties of Context Free Languages.

UNIT-4 **Pushdown Automata:** Definition of Push Down Automata, Languages of Pushdown Automata, Equivalence of PDA's and CFG's, Deterministic PDA

UNIT-5 **Turing Machine:** Introduction to Turing Machine, The Turing machine, Programming Techniques of Turing Machines, Extension to Basic Turing Machine, Turing Machines and Computers, Undecidable Problems about Turing Machines

TEXT AND REFERENCE BOOKS

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "*Introduction to Automata Theory, Languages, and Computation*" 3rd ed., Pearson Education, ISBN: 81-317-1429-2
2. K.L.P. Mishra, N. Chandrasekaran, "*Theory of Computer Science: Automata, Languages and Computation*" 3rd Edition, PHI, ISBN: 978-81-203-2968-3
3. John C Martin, "*Introduction to Languages and the Theory of Computation",* 3rd ed., Tata McGraw Hill, ISBN: 0-07-066048-4

IT 353: Computer Networks

Teaching Scheme		**Evaluation Scheme**	
Lectures	3 Hrs/Week	Test	20 Marks
Tutorials	1 Hr/Week	Teacher Assessment	20 Marks
Total Credits	4	End-Semester Examination	60 Marks

Contact Hours for this subject is 60

Course Outcomes:

1. Experience with protocols and networking concepts
2. : Define key application-layer concepts, including network services required by applications
3. Describe the principle of working of transport layer and implementation of these principles in existing protocols
4. Express working of network layer and routing protocols
5. Explore underlying principles behind video streaming

UNIT-1 **Computer Networks and the Internet**

The Internet, The Network Core Packet Switching Circuit Switching , A Network of Networks, Delay, Loss, and Throughput in Packet-Switched Networks, Overview of Delay in Packet-Switched Networks, Queuing Delay and Packet Loss, End-to-End Delay, Throughput in Computer Networks ,Protocol Layers and Their Service Models Layered Architecture ,Encapsulation, Networks Under Attack

UNIT-2 **Application Layer**

Principles of Network Applications, The Web and HTTP: Overview of HTTP Non-Persistent and Persistent Connections HTTP Message Format ,User-Server Interaction: Cookies 108, Web Caching, The Conditional GET,**File Transfer**: FTP Electronic Mail in the Internet: SMTP ,Comparison with HTTP, Mail Message Format, Mail Access Protocols, **DNS**–The Internet's Directory Service: Services Provided by DNS Overview of How DNS Works, DNS Records and Messages, Peer-to-Peer Applications, P2P File Distribution, Distributed Hash Tables (DHTs), **Network Management** :What Is Network Management? The Infrastructure for Network Management, The Internet-Standard Management Framework, Structure of Management Information: SMI, Management Information Base: MIB, SNMP Protocol Operations and Transport Mappings

UNIT-3 **Transport Layer**

Introduction and Transport-Layer Services: Overview of the Transport Layer in the Internet, Multiplexing and Demultiplexing, **Connectionless Transport-UDP**: UDP Segment Structure, UDP Checksum, **Principles of Reliable Data Transfer**: Building a Reliable Data Transfer Protocol, Pipelined Reliable Data Transfer Protocols, Go-Back-N (GBN), Selective Repeat (SR)

Connection-Oriented Transport: TCP : The TCP Connection, TCP Segment Structure, Round-Trip Time Estimation and Timeout , Reliable Data Transfer, Flow Control, TCP Connection Management **Principles of Congestion Control** :The Causes and the Costs of Congestion, Approaches to Congestion Control TCP Congestion Control

UNIT-4 **The Network Layer**

Introduction Forwarding and Routing: Network Service Models Virtual Circuit and Datagram Networks, Origins of VC and Datagram Networks, **Router:** Input Processing, Switching, Output Processing, The Routing Control Plane

The Internet Protocol (IP): Forwarding and Addressing in the Internet, Datagram Format, IPv4 Addressing, Internet Control Message Protocol (ICMP), IPv6, A Brief Introduction to IP Security **Routing Algorithms**: The Link-State (LS) Routing Algorithm, The Distance-Vector (DV) Routing Algorithm, Hierarchical Routing, **Routing in the Internet** : Intra-AS Routing in the Internet: RIP, Intra-AS Routing in the Internet: OSPF, Inter-AS Routing: BGP, Broadcast and Multicast Routing

UNIT-5 **Multimedia Networking**

Multimedia Networking Applications: Properties of Video, Properties of Audio, Types of Multimedia Network Applications **Streaming Stored Video**: UDP Streaming , HTTP Streaming , Content Distribution Networks, **Case Studies**: Netflix, YouTube **Voice-over-IP**: Limitations of the Best-Effort IP Service, Removing Jitter at the Receiver for Audio, Recovering from Packet Loss, **Case Study**: Internet Telephony with Skype **Protocols for Real-Time Interactive Applications**: RTP 623, SIP **Network Support for Multimedia** : Dimensioning Best-Effort Networks, Providing Multiple Classes of Service

TEXT BOOKS

- “Computer Networking- a top-down approach featuring the Internet”, by James F. Kurose and Keith W. Ross, Person Education, ISBN-10- 0132856204, 6th Edition.
- “Data Communications and Networking”, by Forouzan B. A, Tata McGraw-Hill Publications, 2006, ISBN-0-07-063414-9, 4th edition.

REFERENCE BOOKS

- “Computer Networks”, by Tanenbaum A. S., Pearson Education , 2008, ISBN- 978-81-7758-165-2, 4th Edition
- “Computer Networks and Internet”, by Comer D., Pearson Education, ISBN-81-297- 0330-0, 2nd Edition.
- “Computer Networks- A Systems Approach”, by Larry L. Peterson and Bruce S. Davie, Morgan Kaufmann, ISBN-978-81-312-1045-1, 4th Edition.

IT 355: Operating System

Teaching Scheme		**Evaluation Scheme**	
Lectures	03Hrs/Week	Test	20 Marks
Tutorials	__Hrs/Week	Teacher Assessment	20 Marks
Total Credits	03	End-Semester Examination	60 Marks

Contact Hours for this subject is 45

Course Outcomes:

1. Students will be able to distinguish different styles of operating system design.
2. Students will understand device and I/O management functions in operating systems as part of a uniform device abstraction.
3. Students will have an understanding of disk organization and file system structure.
4. Students will be able to give the rationale for virtual memory abstractions in operating systems.
5. Students will understand the main principles and techniques used to implement processes and threads as well as the different algorithms for process scheduling.
6. Students will understand the main mechanisms used for inter-process communication.
7. Students will understand the main problems related to concurrency and the different synchronization mechanisms available.

UNIT-1 **Introduction** : Operating system and functions, organization of a computer system, operational view of a computing system with resources like processor, memory, input and output, issues in resource management, introduction to the issues in communication with devices, kernel and shell of an operating system ,Classification of Operating systems- Batch, Interactive, Time sharing, Real Time System, Multiprocessor Systems, Multiuser Systems, Multiprocessor Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Monolithic and Microkernel Systems, System calls, Concept of Virtual machine.

UNIT-2 **Process Management**: Process Concept, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process address space, Process identification information, Threads and their management, Scheduling Concepts, Performance Criteria, Scheduling Algorithms, Multiprocessor Scheduling Inter Process Communication models and Schemes, Process generation.

UNIT-3 **Process Management with Synchronization**: Concurrent Processes, Principle of Concurrency, Producer / Consumer Problem, Mutual Exclusion, Critical Section Problem, Peterson’s solution, Semaphores, Test and Set operation; Classical Problem in Concurrency- Dining Philosopher Problem, Monitors, Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock.

UNIT-4 **Memory Management**: Memory management need, memory relocation, linking and loading of memory, processes and primary memory management, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Paging, hardware support for paging, segmentation, segmentation with paging, Paged segmentation, fragmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

UNIT-5 **File System and I/O Management:** File concept, File system structure, file allocation methods (Contiguous, linked and indexed allocation), Disk space management, Swap-space management File sharing, I/O Management and disk scheduling.

Protection & security: Goals of protection, Access matrix, Implementation of access matrix, The security problem, Authentication, Programming threats, System threats, Threat monitoring, Encryption.

TEXT AND REFERENCE BOOKS

1. Abraham Silberschatz and Peter Barer Galvin, “*Operating System concepts”,* 8th Ed. Addison Wesley, 1998
2. Andrew S. Tanenbaum, *"Modern Operating Systems",* Prentice Hall, 1991
3. D M Dhamdhere, “Operating Systems : A Concept based Approach”, 2nd Edition

IT 354 : Advance Database Management Systems

Teaching Scheme		**Evaluation Scheme**	
Lectures	4 Hrs/Week	Test	20 Marks
Tutorials	0 Hrs/Week	Teacher Assessment	20 Marks
Total Credits	4	End-Semester Examination	60 Marks

Contact Hours for this subject is 60

Course Educational Objectives:

1. Identify, describe, and categorize database objects
2. Administer a database by recommending and implementing procedures including database tuning, backup and recovery
3. Propose, implement and maintain database security mechanisms
4. Explore non-relational database systems and structures

Course Outcomes Expected:

1. Design and implement advanced queries using Structured Query Language
2. Design, construct and maintain a database and various database objects using procedural language constructs, forms and reports to solve problems
3. Design and implement a complete problem solution using current database technology. (Oracle 11g)

UNIT-1	PL/SQL – Introduction to PL/SQL – Declare, begin statements, Variables, Control Structure, PL/SQL Transactions – Savepoint, Cursor, PL/SQL Database Objects – Procedures, Functions, Packages, Triggers. Programmatic SQL – Embedded SQL, Dynamic SQL, and ODBC Standard
UNIT-2	Definition of Transaction and ACID properties. Transaction Processing - Transaction-processing monitors, transactional workflows, main-memory databases, real-time transaction systems, long-duration transactions, transaction management in multi-databases. Concurrency Control – Locks, Optimistic Concurrency Control (Backward and Forward validations), Time stamping Concurrency Control.
UNIT-3	**Parallel and Distributed Databases :**Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems- Distributed Systems – Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism – Distributed Database Concepts - Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing – Three Tier Client Server Architecture- Case Studies**.**
UNIT-4	**Object and Object Relational Databases :**Concepts for Object Databases: Object Identity – Object structure – Type Constructors – Encapsulation of Operations – Methods – Persistence – Type and Class Hierarchies – Inheritance – Complex Objects – Object Database Standards, Languages and Design: ODMG Model – ODL – OQL – Object Relational and Extended – Relational Systems : Object Relational features in SQL/Oracle – Case Studies
UNIT-5	**Database security** Security and integrity threats, Defence mechanisms, Statistical database auditing & control. Security issue based on granting/revoking of privileges, Introduction to statistical database security. PL/SQL Security – Locks – Implicit locking, types and levels of locks, explicit locking, Oracles' named Exception Handlers.

TEXT AND REFERENCE BOOKS

1. Silberschatz A., Korth H., and Sudarshan S., DatabaseSystem Concepts, McGraw-Hill (5th Ed), 2006, ISBN: 0072958863
2. Elmasri and Navathe, Fundamentals of Database Systems, Pearson Education
3. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, McGraw-Hill
4. C. J. Date & Longman, Introduction to Database Systems, Pearson Education

IT 356- Elective II Advanced Java

Teaching Scheme		**Evaluation Scheme**	
Lectures	04Hrs/Week	Test	20 Marks
Tutorials	00Hrs/Week	Teacher Assessment	20 Marks
Total Credits	04	End-Semester Examination	60 Marks

Contact Hours for this subject is 60

Educational Objectives:

1. Introduce the students to J2EE.
2. To enable the students to understand the advanced concepts of the Java Language.

Course Outcome Expected:

1 To enable students to develop Network based applications

2 To enable students to develop Advanced Online Applications in Java

UNIT-1 **Introduction to Java 2 Enterprise Edition:**
Need for J2EE, Advantages of J2EE, Types of Enterprise Architecture, Architecture of J2EE, J2EE Components, J2EE Containers, J2EE Technologies

UNIT-2 **Socket Programming:** Introduction to Networking, Client Sockets and Server Sockets
Introduction to RMI: Architecture of RMI, Working with RMI, Creating Distributed Applications, Using RMI, RMI over IIOP, RMI-IIOP and J2EE, Sample Application with RMI-IIOP

UNIT-3 **Servlet Programming:** Overview of Servlet, What's new in Servlet, Features of Java Servlet, Servlet Life cycle,Servlet Configuration, Understanding Request and Response Object, Reading Form Data from Servlet
Understanding Servlet Sessions: What is Session? Introduction to Session Tracking, Mechanism of Session Tracking, Session Tracking and Java Servlet API

UNIT-4 **JSP**: Introduction, Comparison between JSP & servlet., Architecture/Life cycle, Different types of JSP architectures and relative comparison.; JSP tags ,Directives, Scripting elements, Actions; JSP implicit objects, Accessing user information using implicit objects.

UNIT-5 **Enterprise Java Beans:** EJB Fundamentals, EJB Architecture, EJB Interfaces, EJB Roles, Benefits and Limitations of EJB, Session Bean, Stateless versus Stateful Session Beans, Developing Session Beans, Introduction of Entity Beans,BMT Entity Beans, CMP Beans, ejbLoad(), ejbStore() ejbRemove and Finder methods, Sample Application, Deploying EJBs.

TEXT AND REFERENCE BOOKS

Text Books:

1. “Java Server Programming, Black Book”, Dreamtech Press, Edition 2007.
2. “J2EE – Complete Reference”, McGraw Hill, Edition 2007

Reference Books:

1. Bruce Eckel, “Thinking in Java”, Prentice - Hall
2. Herbert Schildt, Patrick Naughton, “JAVA 2 – Complete Reference”, McGraw Hill

IT 357: Object Oriented Modeling and Design

Teaching Scheme		**Evaluation Scheme**	
Lectures	4 Hrs/Week	Test	20 Marks
Tutorials	0 Hrs/Week	Teacher Assessment	20 Marks
Total Credits	4	End-Semester Examination	60 Marks

Contact Hours for this subject is 60

Course Objectives:

1. Release the methodology that encompasses a wide range of software engineering techniques used in system analysis, modeling and design.
2. Integrate well with orientation & importance of knowledge of Object Orientation structural & behavioral modeling techniques.
3. Demonstrate the framework for software engineers to collaborate in the design and development process.
4. Design aspect with UML technology.

Course Learning Outcomes:

1. Ability to build software projects based on Object oriented software engineering practices.
2. Familiar with the UML processes on software development.
3. Industry-readiness: Learner experiments OO methodology to undertake real life software in IT industry.
4. To gets hands on expertise on UML techniques on software Projects.

UNIT-1 **Principles of Object Orientation Motivations for OOP**
Object Oriented development and themes, evidence for usefulness, modeling as a design techniques
Objects, classes, links and associations, generalization and
Inheritance, grouping constructs, aggregation, abstract classes generalization as extension and restriction, multiple inheritance

UNIT-2 **Design Methodology**
Impact of an object oriented approach, Analysis, System design with examples, combining models, Designing models, designing Algorithms, Optimization of design control, Associations, Physical packaging, Comparing methodologies using structure analysis and design, Jackson's structured Development, Information modeling notation and object oriented works

UNIT-3 **OO Programming Languages**
A Comparative Study of some typical Object Oriented Programming Languages such as C++, JAVA.
Structural Modeling using UML
Classes, Relationships, Common mechanisms, Diagrams, Class Diagrams, Interfaces, Types and Roles, Packages, Instances and Object Diagram

UNIT-4 **Behavioral Modeling using UML**
Interactions, Use cases, Use case diagram, Interaction Diagrams and Activity diagrams, Events and signals, State Machines, Processes and Threads, Time and space, State chart diagrams

UNIT-5 **Architectural Modeling using UML**
Components, Deployment, Collaboration, Patterns and Frame works, Component Diagrams, Deployment Diagrams

Textbook:

1. Rambaugh, Premerlani, Eddy, Lorenson, *"Object Oriented Modeling and Design"*, ,PHI.
2. Grady Booch, Jeams Rambaugh, IvarJacotson, *"The Unified Modeling Language User Guide",* Addison Wesley

References:

1. Andrew High, *"Object Oriented Analysis and Design",* TMG
2. Kahate, *"Practical Object Oriented Design with UML",* Mark Priestley.

IT 358: Lab: Computer Networks

Teaching Scheme		**Evaluation Scheme**	
Practical	2 Hrs/Week	Term Work	25 Marks
Credits	1	Practical/Viva-voce	25 Marks

Contact Hours for this subject is 15

Course Outcomes:
CO1: Demonstrate routing algorithm
CO2: Exercise socket programming
CO3: Demonstrate sliding window protocols
CO4: Examine packets

Suggested list of experiment:

1 Write a program for distance vector algorithm to find suitable path for transmission.

2 Using TCP/IP sockets, write a client-server program file transfer

3 Write a program for congestion control using Leaky bucket algorithm.

4 Write a program for implementation of simple DNS

5 Write a program for to simulate a sliding window protocol: Go Back N

6 Write a program for to simulate a sliding window protocol: Selective Repeat

7 Study of Network Simulation
   a. Platform required to run network simulator
   b. Backend Environment of Network Simulator
   c. Basics of Tcl Programming for NS-2

8 Simulating a Local Area Network (in virtual Labs)
   a. Study of Ethernet Frame Structure
   b. Simulating a LAN using Network Simulator 2

9 Study of Wireshark
   a. Platform required
   b. Installation
   c. Packet sniffing and analysis using Wireshark

10 Study of Tcpdump

IT 359 : Lab-Advance Database Management Systems

Teaching Scheme		**Evaluation Scheme**	
Practical	2 Hrs/Week	Term Work	25 Marks
Credits	1	Practical/Viva-voce	25

Contact Hours for this subject is 15

Course Outcomes Expected

1. Implement SQL DDL and DML commands
2. Implement SQL functions and procedures
3. Design and implement SQL forms and Reports
4. Design and implement a complete problem solution using current database technology. (Oracle 11g)

The term work shall consist of following practicals

1. SQL
1. Simple Queries using DDL, DML and DCL
2. SQL Aggregate Functions
3. SET Operations
4. Views and Snapshots
5. Multiple Tables and Nested Queries PL/SQL
6. PL/SQL Block
7. Function and Procedures
8. Subprograms and Packages
9. Triggers
10. Cursors

FORMS AND REPORTS
11. Designing Oracle Forms using Menus and Buttons
12. Developing Oracle Reports

TERM Work:
The term work consists of at least 10 experiments/ assignments based on the syllabus of the subject.
Assessment of term work should be done based on the points mentioned below:

- Continuous lab assessment
- Actually performing practices in the laboratory
- Oral Evaluation conducted (internally) at the time of submission

IT 360: Lab Software Development Lab II (ASP.NET)

Teaching Scheme		**Evaluation Scheme**	
Practical	02Hrs/Week	Term Work	25 Marks
Credits	01	Practical/Viva-voce	25 Marks

Contact Hours for this subject is 15

Course Outcomes:

1. Design secure web applications using ASP.Net
2. Create, and test web service procedure using ASP.Net and IIS.
3. Implement a simple web service to demonstrate the use of rich controls for creating online registration form
4. Develop Windows forms and data driven applications using various controls.

The term work shall consist of following practicals

1 To Study the ASP.Net Introduction.
2 Procedure to set up the ASP.Net and IIS.
3 Create a program for Online Test using ASP.NET
4 Design logon Web form and validate it
5 Demonstrate the rich controls for creating online registration form
6 Save logon and registration information in Database.
7 Develop a web page to insert, delete & modify information stored in the database.
8 To create advertisements using adRotator
9 Create a simple web service
10 Mini Project

IT 361: Lab- Operating System

Teaching Scheme		**Evaluation Scheme**	
Practical	02Hrs/Week	Term Work	25 Marks
Credits	01	Practical/Viva-voce	25 Marks

Contact Hours for this subject is 15

Course Outcomes:
1. Students will be able to discuss the characteristics of different structures of the Operating Systems (such as microkernel, layered, virtualization, etc.) and identify the core functions of the Operating Systems.
2. Students will be able to explain the principles and compare the algorithms on which the core functions of the Operating Systems are built on.
3. Students will be able to analyze and evaluate the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions.
4. Students will be able to demonstrate knowledge in applying system software and tools available in modern operating system (such as threads, system calls, semaphores, etc.) for software development.

The term work shall consist of following practical/assignments/mini-project/visit*

1 Installation Process of various operating systems and implement various UNIX commands

2 Shell Programming: Basic of shell programming with implementation of Examples

3 Write a program using system calls of UNIX OS (fork,exec,wait)

4 Write a Program to simulate process scheduling like FCFS, Shortest Job First and Round Robin

5. Write a program for interposes communication using Shared memory
6. Write a program for interposes communication using pipes
7. Write a Program for deadlock avoidance(Bankers algorithm)
8. Write a Program to simulate page replacement algorithms like FIFO and LRU.
9. Write a Program to simulate memory allocation using First fit, Best fit, and Worst fit.
10. Case study of latest operating system

References:

Unix concepts and applications by Sumitabha Das, TMH Publications.
Unix programming by Stevens, Pearson Education.
Shell programming by Yashwanth Kanetkar.
Operating System Concepts by Silberschatz, and Peter Galvin.

IT 362: Lab Advanced Java

Teaching Scheme		**Examination Scheme**	
Practical	2Hrs/Week	Term work	25 Marks
Credits	1	Practical/Viva	25 Marks

Course Outcomes Expected:

1 Demonstrate JEE.
2 Implement Networking using the concept of Socket programming and RMI.
3 Write programs using Servlet, JSP,EJB.

Contact Hours for this subject is 15

TERM Work:

The term work shall consist of a journal record of at least 10 experiments/ assignments
Assessment of term work should be done based on the points mentioned below:
- Continuous lab assessment
-Actually performing practical in the laboratory
-Oral Examination conducted (internally) at the time of submission

Suggestive List of experiments:

1. Program to demonstrate simple chat application using Networking.
2. Program to develop RMI application for basic arithmetic calculations.
3. Create a Servlet to read data from a HTML From and display it on Servlet page.
4. Create a Cookie using Servlet API.
5. Write a JSP to output the values returned by Systems.getProperty for various system properties such as java.version, java.home, os.name, user.name, user.home, user.dir etc.
6. Create a sample application using JSP.
7. Create a Simple EJB to demonstrate Java EJB API.
8. Deploy Bean created in Previous Experiment.
9. Mini Project

Practical Examination:

The Practical Examination shall consist of writing and performing an experiment / assignment and oral based on the syllabus as per the journal record. Duration of examination is three hours.

IT 363: Lab Object Oriented Modeling and Design

Teaching Scheme		**Evaluation Scheme**	
Practical	02Hrs/Week	Term Work	25 Marks
Credits	01	Practical/Viva-voce	25 Marks

Contact Hours for this subject is 15

TERM Work:

The term work shall consist of a journal record of at least 10 experiments/ assignments
Assessment of term work should be done based on the points mentioned below:
- Continuous lab assessment
-Actually performing practical in the laboratory
-Oral Examination conducted (internally) at the time of submission

Course Learning Outcomes:

1. Ability to build software projects.
2. Familiar with the UML processes on software development.
3. Ability to undertake and develop real life software for IT industry.
4. To gets expertise on UML techniques on software Projects.

1 To develop a problem statement.
2 Develop an IEEE standard SRS document. Also develop risk management and Project plan (Gantt chart).
3 Identify Use Cases and develop the Use Case model.
4 Identify the business activities and develop an UML Activity diagram.
5 Identity the conceptual classes and develop a domain model with UML Class diagram.
6 Using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.
7 Draw the State Chart diagram.
8 User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
9 Implement the Technical services layer.
10 Implement the Domain objects layer.
11 Implement the User Interface layer.
12 Draw Component and Deployment diagrams.
13 Mini Project

Suggested Software Tools

ArgoUML, Eclipse IDE, Visual Paradigm, Visual case, and Rational Suite

Practical Examination:

The Practical Examination shall consist of writing and performing an experiment / assignment and oral based on the syllabus as per the journal record. Duration of examination is three hours.