BSc (Computer Science course structure) **Old Course**

BSc Semester I		Periods	Marks			Grand Total
			CIA	ESE	PRA	
CS 1	Introduction to Problem Solving	3T + 3P	15	60	25	100
CS 2	Computer Architecture and Organization	3T + 3P	15	60	25	100
BSc Semester II						
CS 3	Data Structures	3T + 3P	15	60	25	100
CS 4	Operating Systems	3T + 3P	15	60	25	100
BSc Semester III						
CS 5	Object Oriented Programming	3T + 3P	15	60	25	100
CS 6	DBMS – I	3T + 3P	15	60	25	100
BSc Semester IV						
CS 7	Client Side Web Development	3T + 3P	15	60	25	100
CS 8	DBMS – II	3T + 3P	15	60	25	100
BSc Semester V						
CS 1	Computer Networks	4T	20	80	-	100
CS 2	Android development	4T	20	80	-	100
CS 3	Human Computer Interface	4T	20	80	-	100
CS 4	Agile Software Development	4T	20	80	-	100
PL1	Practical Lab I	8P	-	-	100	100
PL2	Practical Lab II	8P	-	-	100	100
PR	PROJECT	2T	-	-	-	-
BSc Semester VI						
CS 5	Network Security	4T	20	80	-	100
CS 6	Multimedia Techniques	4T	20	80	-	100
CS 7	Server Side Web Development	4T	20	80	-	100
CS 8	Embedded Systems	4T	20	80	-	100
PL3	Practical Lab I	8P	-	-	100	100
PL4	Practical Lab II	8P	-	-	100	100
PR	PROJECT	2T	50	50	-	100

Foundation Course :

Semester III : Introduction to Cyber security and cyber law

Semester IV : Foundation Course : Information Systems & IT Entrepreneurship

FY BSC Semester I

CS 1: Introduction to Problem Solving

Learning Objective: To make the student understand the concept of computer algorithm & use the algorithm for various problem solving.

Learning Outcomes: By the end of this paper, the student will be able to list steps involved in problem solving, describe basic problem solving strategies, determine the best strategy for solving basic mathematical problems, design flowchart for basic mathematical problems, and model a flowchart into a C program.

Theory(75 marks) Lectures: 45

1. Algorithm: Definition, specification & pseudocode conversion, recursive algorithms. [3L]
2. Elementary algorithms: summation of a set of numbers, factorial computation, generation of the Fibonacci sequence, reversing the digits of an integer, base conversion. [6L]
3. Factoring method: finding square root of a number, the smallest divisor of an integer, the greatest common divisor of two integers, generating prime numbers, computing prime factors of an integer. [6L]
4. Array techniques: array order reversal, array counting or histogramming, finding maximum of a set, removal of duplicates from an ordered array, finding the k-th smallest element. [6L]
5. Merging, sorting, and searching: two-way merge; selection sort, insertion sort; binary search, hashing. [12L]
6. Recursive algorithms: binary tree traversal; recursive algorithm; towers of Hanoi problem, combination generation, permutation generation. [6L]
7. Problem solving aspects such as definition of problem, getting started on a problem, general problem solving strategies; top-down approach; implementation of algorithms; efficiency of algorithms. [6L]

Text Book:

1. RG Dromey, How to solve it computer, PHI

References:

1. Aho, Hopcropt, Ullman, The design and analysis of computer algorithms, Pearson Education
2. Rosen Kenneth, Discrete mathematics and its applications, fifth edition, Tata McGraw-Hill
3. E Horowith, S Sahni, S Rajasekaran, Fundamentals of computer algorithm, Galgotia.

Practicals (25 Marks)

Lab assignments can be done using C or Python. It is mandatory to use Flow charting, algorithm using pseudo language, tracing of algorithms. A suggested list of assignments can be:

1. Elementary algorithms

2. Factoring method
3. Array techniques
4. Merging
5. Sorting
6. Searching
7. Recursive algorithms

Note: Tools like scratch can be used to supplement the practical.

CS 02: Computer Architecture and Organization

Learning Objective: To understand the building blocks of computer and study various design issues. To be able to design, execute and debug assembly language programs.

Learning Outcomes: The student will be able to outline components of CPU, memory system, ALU, control unit, superscalar processors, describe types of memory, I/O communication, operation of control unit, RISC architecture.

Theory(75 marks) Lectures 45

1. COMPUTER SYSTEM: Computer components, Computer functions interconnection structure- CPU, Memory, I/O, Bus interconnection. [3L]
2. INTERNAL MEMORY: Characteristics of memory system, the memory hierarchy, semiconductor main memory, memory organization, Memory interleaving Cache memory- principles, elements of cache design, mapping functions, replacement algorithms, write policy, block size. [8L]
3. INPUT/OUTPUT: External devices, I/O modules, Programmed I/O, Interrupt driven I/O, DMA, I/O channels and processors. [6L]
4. COMPUTER ARITHMETIC: number systems, ALU, integer representation, integer arithmetic, floating-point representation and arithmetic [6L]
5. INSTRUCTION SETS: Machine instructions characteristics, Types of operations-data transfer, arithmetic, logical, conversion, I/O, system control, transfer of control; Assembly language, Addressing modes-immediate, direct, indirect, register, register indirect, displacement, stack; Instruction formats- instruction length, allocation of bits, variable length instructions. [9L]
6. CPU: Processor organization, registers organization, instruction cycle-fetch, execute, interrupt, indirect data flow. [5L]
7. CONTROL UNIT: Micro operations, control of CPU-functional requirements, control signals, internal CPU organization, and hardwired implementation, micro-programmed control unit, advantages of micro-programmed control unit, merits and demerits over hardwired control unit [6L]
8. *Advanced topics: RISC concepts, superscalar processors, parallel processing, Basic concepts of pipelining, advantages and disadvantages [2L]

*- Brief idea only.

Text book:

1. William Stallings, Computer Organization and Architecture-Designing for performance (Fourth Edition) by EEE, PHI. (Seventh edition in market)
2. Carl Hamachar, Zvonco Vranesic and Safwat Zaky, Computer Organization, McGraw Hill

Practicals (25 marks)
Suggested list of practical is given below: C programs to perform binary and hexadecimal arithmetic(20%) , Introduction to 8086 architecture and instruction set and Writing assembly language programs in 8086 (80%) using MASM or compatible assembler either in windows or linux.

Semester II
CS3: Data Structures

Learning Objective: The objective of the course is to introduce the fundamentals of Data Structures, Abstract concepts and how these concepts are useful in problem solving. This course aims at the student to be able to â€“

- Understand and use the process of abstraction.
- Analyze step by step and develop algorithms to solve real world problems.
- Implement various data structures viz. Stacks, Queues, Linked lists, Trees.
- Understand various techniques of searching and sorting.

Learning Outcomes: By the end of this paper, the student will be able to define data structures, explain the applications of each data structure, apply data structures for real life problems, analyze real life problems for data and functionality, compare the efficiencies of different ways of solving problems, design recursive functions for special real life problems, and derive data structures and functions for real life problems using C programming.

Theory(75 marks) Lectures 45

1. Introduction to data structure : Basic concept of data, problem analysis, algorithm complexity, Big O notation, Big-omega, Big-theta notations and time space trade-off [3L]
2. Stacks and Queues : Basic of stacks and queues, Recursion, Polish notation, circular queues and priority queues. [5L]
3. Linked list : Single linked list, circular linked list, doubly linked list and dynamic storage management, generalized list, garbage collection. [10L]
4. Trees : Definition & concepts, basic trees, binary tree representations, binary tree traversals and applications of trees & binary search tree, expression tree. [6L]
5. Graphs : Introduction, application and representation. Shortest path algorithm. [4L]
6. Searching and Sorting : use of various data structures for searching and sorting, linear and binary search, insertion sort, selection sort, bubble sort, merge sort, quick sort and heap sort. [12L]

Text Book:

1. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures : A Pseudocode Approach with C, Cengage Learning, 2 Edition (Paperback), 2007.

References:

1. Ellis Horowitz, Sartaj Sahni, Fundamentals of Data Structures in C, Universities Press, 2nd Edition, 2008.

Practical: (25 marks)

Lab assignments can be done using C or Python. Use of make and debug tool to be preferred with multi-file development. Topics to be covered in laboratory are as follows:

- Recursion
- Stack and Queue
- Linked list
- binary tree
- Sorting: insertion sort, selection sort, bubble sort, merge sort, quick sort and heap sort
- Searching: Linear and Binary

CS4: Operating Systems

Learning Objective: To study the modern day operating systems with emphasis on its functions and structure so as to enable students to decide the suitable operating system for specific job.

Learning Outcomes: The structure and functions of operating system are understood by the students. The students are able to choose the appropriate operating systems according to the type and need of the software to be deployed.

Theory(75 marks) Lectures 45

1. Introduction to Operating System: Basic elements of a computer system: Processor, Main Memory, I/O Modules, System Bus, Instruction Execution; Operating Systems: Definition, Evolution, Introduction to Major Functions/Services, OS Structure, Relationship between Kernel, OS, Hardware [4L]
2. Processes & Process Management: Process: Definition, Process Control Block, Process States, Operations on Process; Threads and Microkernels: Definition, Multi-threading Model; Process Scheduling, Introduction to the Concept, Scheduling Criteria, Scheduling Algorithms, Multi-Processor Scheduling; Concurrency/ Process Coordination, Synchronization, Principles, Mutual Exclusion, The Critical-Section Problem, Petersons Solution, Semaphores, Monitors, Readers/Writers Problem; Deadlock, Principles, Deadlock Handling Methods, Prevention, Avoidance, Detection, Recovery From Deadlock [12L]
3. Memory Management: Memory Management Concepts, Introduction, Swapping, Contiguous Memory Allocation, Paging, Page Table, Segmentation; Virtual Memory, Introduction, Demand Paging, Page Replacement, Frames, Thrashing [10L]
4. Input/ Output & File System: File System, Concepts, File Organization and Access Methods, Directory Structure, File Sharing; I/O Management, I/O devices, I/O Hardware, Organization of I/O, I/O Buffering, Disk Structure, Attachment, Scheduling and

Management, RAID [12L]

5. Advanced Concepts: Distributed Operating System, Reasons for Distributed OS, Types, Design Issues, File Systems on Distributed OS, Synchronization (Introduction); Web Based Operating Systems, Types, Advantages, Storage Structure, Resource management [7L]

Text Book:

1. Silberchatz Galvin; OS Principles; (7e)

Reference Book:

1. William Stallings; Operating Systems; (5e)

Practical: (25 marks)

Suggested list of assignments:

1. Installing Linux / Windows Operating System, Partitioning and formatting disk, Installing applications device drivers, working with files, mounting file systems, checking system space, creating, modifying and deleting user accounts
2. Study of Basic commands of Linux.
3. Study of Advance commands of Linux.
4. Shell Programming in Unix/Linux, arithmetic operations, loops, files

   Ex. Write a BASH shell script prime which will accept a number b and display first n prime numbers in standard output.
5. Shell scripting using general-purpose utilities.

   Ex. A) Write a menu driven shell script which will print the following menu and execute the given task to display result on standard output.

   1. Display calendar of current month
   2. Display today’s date and time
   3. Display usernames those are currently logged in the system
   4. Display your name at given x, y position
   5. Display your terminal number
   6. Exit
6. Shell programming using filters (including grep, egrep, fgrep)
7. Write a shell script to validate the entered date. (eg. Date format is : dd-mm-yyyy)
8. Write a shell script to check entered string is palindrome or not.

Semester III

CS 5: Object Oriented Programming

Learning Objective: Give sufficient understanding of what is the difference in procedural and object oriented way of handling given problem

Learning Outcomes:

Students will be able to use object oriented concept while programming which include following

- To define data and objects.
- To understand need to use Object Oriented concepts in programming.
- To understand the problems with procedure oriented programming.
- To define following
    - Encapsulation
    - Data Abstraction
    - Data Hiding
    - Abstract data types
- Recall and use data types and data structure statements while writing programs.
- Analyze which data structure will be best to provide solution for appropriate problem.
- Test the write code to get correct output and optimized code.
- To define syntax Abstract data types with a class, objects, members, controlling access to members, packages, Interfaces, initializing class objects using constructors, overloaded constructors, finalizers, Set and Get methods, friendly access(package access) composition-objects as instance variables of other classes
- To write the program with Abstract data types with a class, object, members
- To define Collections, Threading, Serialization, Generic programming.
- To use Collections, Threading, Serialization, Generic programming

Theory(75 marks) Lectures 45

1. Should we think of Data or Objects? problems with procedure oriented programming, other ways of simulating the world scenarios, Case study of a Savings bank account. [2L]
2. Introduction to Object Oriented Programming, Reusability and Extensibility, abstract data type, Encapsulation, data abstraction and data hiding. [3L]
3. Introduction to: variables, data types, operators, control structures, arrays, methods, method overloading and method overriding. [2L]
4. Object based programming : Introduction, Abstract data types with a class, objects, members, controlling access to members, packages, Interfaces, initializing class objects using constructors, overloaded constructors, finalizers, Set and Get methods, friendly access(package access) composition-objects as instance variables of other classes [12L]
5. Using this reference, static class members, Inheritance, superclasses and subclasses, Internal Class, protected members, using constructors and finalizers in subclass, composition v/s inheritance, polymorphism, dynamic method binding, final methods and classes, abstract superclass and concrete class, Interfaces, packages, String operations. [12L]
6. Exception handling- Try, throw and catch blocks, rethrowing an exception, throws clause,

finally block, exception handling in constructors and inheritance. [4L]

7. Collections, Threading, Serialization, Generic programming [10L]

Text book:

1. Timothy A. Budd, Introduction to Object-Oriented Programming, 3^{RD} edition, Pearson Education India

References:

1. Mahesh P. Matha; Core Java : a Comprehensive Study; PHI Learning Pvt. Ltd., 2011
2. http://en.wikibooks.org/wiki/Java_Programming

Practical: (25 marks)

List of assignments:

Simple programs covering various features of Java: Datatypes, operators, Control structures, Arrays, Methods Concepts of OOP, Simple Class, Controlling access to members Constructors, Composition, Inheritance , superclass, subclass and protected members Polymorphism, Exception Handling, collections

CS 6 : DBMS

Learning Objective: To make aware of how to design and organise the data for easy manipulation or access.

Learning Outcomes:

The students will be able to:

- Explain database concepts, technology and practice.
- Formulate SQL statements and queries using SQL programming.

Theory(75 marks) Lectures 45

1. Introduction to Data Base Systems: File Systems versus a DBMS, The Relational Model, Levels of abstraction in a DBMS, Data independence, Queries in DBMS, Concurrent Access and Crash Recovery, Structure of DBMS, Advantage of DBMS, People who deal with Databases. [4L]
2. Conceptual design and Entity Relationship model: Overview of Data Base Design, The ER model-features, Key Constraints, Participation Constraints, weak Entities, Class Hierarchies, Aggregation, Entity versus attribute, Entity versus relationship, Binary versus ternary relationship, aggregation versus ternary relationships. [6L]
3. The Relational Model: Attributes and domains, Relations, Integrity Constraints, Key Constraints, Foreign Key Constraints, General Constraints, Enforcing Integrity constraints. [3L]
4. Logical Database design ER to relational : Entity sets to tables, Relationship sets (without constraints) to tables, translating relationship sets with key constraints, translating relationship sets with participation constraints, translating weak entity sets, translating class hierarchies, translating ER diagrams with aggregation. [10L]
5. Schema Refinement and Normal forms: Introduction, Why Schema Refinement? Functional

Dependencies, Normal Forms: BCNF, Third Normal Form, Normalisation-Decomposition upto BCNF[7L]

6. SQL: The Form of Basic SQL query, Condition specification, SQL Join, Union, Intersect, Except, Nested queries, Aggregate Operators, Null values. [15L]

Practical: (25 marks)

Practicals should be done using a DBMS software like Oracle, SQL Server, MYSQL

1. Gathering information, Analyzing data, ER Diagram, Reduction to Tables.
2. Creation/modification of database tables using DDL statements and GUI tools of the DBMS software.
3. SQL queries
4. SQL Joins, Views.
5. User management, granting/revoking privileges, roles.
6. Report generation

Note : A minimum of 5 example sets covering all concepts should be done for topics under Serial No 1(ERDs), 3 (SQL).

Semester IV

CS 7 : Client Side Web Development

Learning Objective: To develop a high degree of competence as a web developer by learning principles and techniques of client-side programming with HTML, CSS and Javascript. The student will also learn how to secure it’s e-existence and report the cyber crimes if any.

Learning Outcomes:

The students will be able to:

- To develop a high degree of competence as a web developer by learning principles and techniques of client-side programming with HTML, CSS and Javascript.
- The student will be able to secure it’s e-existence and report the cyber crimes if any.

Theory(75 marks) **Lectures 45**

1. Basic HTML: [9L]

   Introduction, importance, Doctype, Header, Body and footer of HTML page, Techniques of HTML: Lists, Table, Imagemap, Hyperlinks, Form, Frames.

2. CSS: [9L]

   Introduction to CSS, Advantages, Types of style sheets: Inline, Internal, External, Multiple Style sheets and Cascading order, Grouping or nesting, Syntax, ID and Class, Pseudo-class, Pseudo-element, CSS units of measurement, Colors.

3. Client side scripting - Javascript and Ajax: [6L]

   DOM, Introduction to Javascript, Functions and variables, introduction and use of Ajax.

4. XML: [12L]

   History of XML, XML Basics, syntax, XML Schemas, Document Type Definitions(DTD), XML Namespaces, XSL(Extensible Style Sheet Language), Generating XML from database, using XSL to format XML

5. Trends in Web developments: [9L]

   What's new in HTML5, CSS3; Introduction to Framework.

Text Book:

1. Elisabeth Robson, Eric Freeman, Head First HTML with CSS & XHTML A Learner's Companion to HTML, CSS and XHTML, O'Reilly Media

Practical: (25 marks)

Basic: HTML Tag, HEADER Tag, META Tag, TITLE Tag, BODY Tag, Text Formatting : PRE Tag, FONT Tag, entities, Image: IMG tag, image maps, Hyperlinks : Anchor tag, Lists : Unordered Lists, Ordered Lists, Definition Lists, Table tags : TABLE, TR and TD Tags, Cell Spacing and Cell Padding, Colspan and Rowspan, Frames: Frameset, FRAME Tag, NOFRAMES Tag, Enduser Interaction : FORM and INPUT Tag, Text Box, Radio Button, Checkbox, SELECT Tag and Pull Down Lists, TEXTAREA, Hidden, Submit and Reset, Special Tags : COLGROUP, THREAD, TBODY, TFOOT, _blank, _self, _parent, _top, IFRAME, LABEL, Attribute for <SELECT>.

Use of Webserver to be done for uploading and fetching the static webpages created. Following can be the sample list of practical assignments:

- Basic and Text formating tags, Lists
- Image, Table, Special Tag
- Frameset, Iframe, Form
- CSS Internal, Inline, External
- Javascript
- Create XML with DTD and XSD
- XML display using XSL

CS 8: DBMS - II

Learning Objective: To know the basics of design and security of databases.

Learning Outcomes: The students will be able to learn basic design and security of database.

Theory(75 marks) **Lectures 45**

1. Physical Database design: File organisation – sequential, heap, hash; variable and fixed length records, Ordered and unordered index, B^+ and B tree index, hashing index. [5L]
2. Relational Algebra & Query processing : Relational algebra operations- select, project, join, natural join,equijoin and their implementation. Basic query processing plan generation [5 L]
3. SQL: Embedded SQL, Cursors, Dynamic SQL, Triggers and active databases [8 L]
4. Concurrency Control : The concept of transaction, transaction and schedule, Notion of consistency, Serializability, Lock based concurrency control, concurrency control without

locking, deadlocks [6L]

5. Crash Recovery: Introduction to crash recovery, Recovery and atomicity, Log based recovery, Shadow paging. [6L]
6. Security : Introduction to Database security & Integrity threats, Defence Mechanisms-views, Access control, discretionary access control, mandatory access control, statistical databases, auditing and control [7L]
7. SQL3 Object relational features [5L]
8. Current trends in database computing: NoSQL, NuoDB, distributed data using clustering, data mining, data warehousing [3L]

Text Book:

1. Silberschatz, Korth, Sudarshan, Database System Concepts, McGraw Hill International Edition

References:

1. Ramez Elmasri, Shamkant Navathe; Fundamentals of Database Systems, Pearson Education
2. Raghu Ramakrishnan, "DataBase Management Systems", McGraw Hill publication

3. Bipin Desai, An Introduction to Database systems, Galgotia Publications

Practical: (25 marks)

Practicals should be done using a RDBMS software like Oracle, SQL Server, MYSQL and front-end like VB.NET / C#

List of suggested Practical:

Normalisation, Stored Procedures, Triggers

Creating forms for data entry

Reports

Use of common dot net controls and event driven programming

Creating a menu system using menu items, listview, mdiforms

Simple Data entry form connecting to a database and inserting records using ADO.NET

Fetching records from the tables

Semester V

CS 1: Computer Networks

Learning Objective: To understand how computers are networked and how they communicate.

Learning Outcome: The students will be able to:

- Define various layer in Network.
- Draw the diagram in Networking.
- Classify the different layers in TCP & OSI model.
- Explain and classify the v arious functions in each of the layer.
- Conclude the functions of each network layer.
- Define the use of various devices in the network layer.

Theory(100 marks) Lectures 50

1. Data Communications: Beginnings of Networking and data communication, ARPAnet; Networks: Components and Categories, Types of Connections, Topologies; Protocols and Standards: Layered Architecture, ISO / OSI model, TCP/IP model; Applications of Networks; Examples of Network [8L]
2. Physical layer: Functions of Physical layer; Data Encoding: Manchester, Differential Manchester; Transmission Media: Twisted pair, Coaxial Cable, Fiber Optics, Wireless Media; Physical layer Devices: Repeaters [8L]
3. Data Link Layer: Functions of Data link layer; Data Framing techniques, Character Count, Character Stuffing, Bit Stuffing; Error detection and correction: Parity, CRC, Hamming code; Protocols: Stop and wait, Go back-N ARQ, Selective repeat ARQ, Sliding window, HDLC; Network Standards: Ethernet IEEE 802.3, , IEEE 802.11, Data Link layer devices: Bridges [8L]
4. Network layer: Functions of Network layer; Network Service types: Virtual Circuits, Datagrams; Routing Algorithms: Shortest path routing, Distance Vector routing, Link State routing; Internetworking; Internet Protocol: Frame Format, Addressing, Subnetting; Network layer devices: Gateways [9L]
5. Transport layer: Functions of Transport layer; Transport Service: Connection less, Connection oriented; Protocols: User Datagram Protocol, Transmission Control Protocol; Quality of Services parameters; [9L]
6. Application layer : Functions of Applications layer; Protocols: FTP, SMTP; Domain Name System; [8L]

Text Book:

1. Behrous A. Forouzan; Data Communications and Networking

References:

1. Andrew S. Tanenbaum; Computer Networks

PL 1: Computer Networks and Android Development

Practical Lab(100 marks)

Group A : Networking Lab

1. Configuring TCP/IP on a desktop,(use of command ipconfig)
2. Using diagnostic Network Commands : ping, traceroute, netstat, nslookup
3. Simulation of framing using Bit stuffing and character stuffing.
4. Simulation of stop and wait protocol.
5. Setting up of wireless network (ad-hoc and Infrastructure)
6. Use of route command.
7. Configuring a modem for Internet connectivity.
8. Implementation of shortest path algorithm.
9. IP address manipulation (Extract network id and Host id given netmask)
10. Simulating IP fragmentation and reassembly.

CS 2: Android development

Learning Objective: Impart knowledge to understand fundamentals of android and create and publish Apps for Google Android devices.

Learning Outcomes: The student will be able to remember components of an app, understand procedure for app development and use them to create new apps.

Theory(100 marks) **Lectures 50**

1. Introduction to mobile apps

   Why we Need Mobile App, Different Kinds of Mobile Apps, Introduction to Android, Pre-requisites to learn Android, Brief discussion on Java Programming

2. Android Architecture overview

   Overview of android stack, Introduction to OS layers, Android features. Linux Kernel, Libraries, Android Runtime, Application Framework, Dalvik VM

3. Android Components

   Activities, Services, Broadcast Receivers, Content Providers

4. Building UI with Activities

   Activities, Views, layouts and Common UI components, Creating UI through code and XML, Activity lifecycle, Intents, Communicating data among Activities

5. Advanced UI

   Selection components (GridView, ListView, Spinner), Adapters, Custom Adapters, Menus, Toast, Custom Toast,Dialogs, Status bar Notifications

6. Multithreading

   Using Java Mutithreading classes, AsyncTask, Handler, Post

7. Intent, Intent Filters and Broadcast Receivers

   Role of filters, Intent-matching rules, Filters in your manifest, Filters in dynamic Broadcast Receivers, Creating Broadcast receiver, Receiving System Broadcast, Understanding Broadcast action, category and data, Sending Broadcast

8. Data Storage

   Shared Preferences, Android File System, Internal storage, External storage.

   SQLite - IntroducingSQLite, SQLiteOpenHelper and creating a database, Opening and closing a database, Working with cursors Inserts, updates, and deletes

9. Content Providers

   Accessing built in content providers, Content provider MIME types, Searching for content, Adding, changing, and removing content, Creating content provider, Working with content files

10. Services

Overview of services in Android, Implementing a Service, Service lifecycle, Inter Process Communication (AIDL Services).

Web Services and WebView - Consuming web services, Receiving HTTP Response (XML, JSON), Parsing JSON and XML, Using WebView.

Text books:

1. Pro Android 4 ; Satya Komateneni, Dave MacLean (Apress)
2. Beginning Android 4 Development, Wei-Ming Lee(John Wiley & Sons)
3. Hello Android - Introducing Google's Mobile Development platform - Ed Brunette (The Pragmetic Bookshelf)
4. Android Apps with Eclipse 1st Edition, Onur Cinar(Apress)
5. Android- A Programmer'S Guide, Dimarzio, J.F.(Tata McGraw Hill)

Web References:

1. http://developer.android.com/index.html
2. http://developer.android.com/design/index.html
3. http://www.appinventor.org/
4. venturesity.com/blog/list-of-free-android-app-development-tutorials
5. http://www.eclipse.org/
6. http://www.smashingmagazine.com/guidelines-for-mobile-web-development/

PL 1: **Group B : Android Development**

Following is a suggested list of practicals:

1. Installing Android Machine - Configuring Android Stack, Creating Eclipse Environment, Integrating Android with Eclipse IDE, Exploring Eclipse IDE
2. Creating First Android Application - Creating Android Project, Debugging Application through DDMS. setting up environment. AVD Creation, Executing Project on Android Screen
3. Android application development - Use of GU components
4. Implementing an animated game - USe multithreading concept
5. Implementing Data storage application - Use SQLite & MySQL
6. Create an application that will pass some number to the next screen , and on the next screen that number of items should be display in the list
7. Understanding content providers and permissions: Read phonebook contacts using content providers and display in list.
8. Create an application to make Insert, update, Delete and retrieve operation on the database.

Note: Use Eclipse for Android application development

CS 3: Human Computer Interface

Learning Objective: To understand the information systems, ethics and issues in them, and how to secure the information systems.

Learning Outcomes: By the end of this course, student will be able to list the principles of good interface design, understand the importance of interactive systems, describe the layers of user interface, determine the best strategy for designing effective user interfaces, design prototypes for user interfaces, evaluate the interface designs based on Neilsons heuristics and create effective user interfaces.

Theory(100 marks) Lectures 50

1. Introduction: Human-Computer Interaction, Evaluating Designs, The Birth of HCI. Importance of user Interface, Importance of good design, Benefits of good design, principles & heuristics of good design [5L]
2. Human interaction with computers, Importance of : Human characteristics, Human consideration, Human interaction speeds, Understanding business functions. User centred design- Need-finding: Participant Observation, Interviewing, Additional Need finding, contextual inquiry & persona.
3. Rapid Prototyping: story boarding. Paper Prototyping and Mockup, Video Prototyping, Creating and Comparing Alternatives
4. Direct Manipulation and Representations: various user interaction models- command, menu, Direct Manipulation. Mental Models. Hueristics(guidelines) for design.
5. Graphical Interface Design: Graphical user interface, standards such as Microsoft windows HCI guidelines, Windows: Navigation schemes selection of window; Selection of devices based and screen based controls, Components, Text and messages, Icons, Multimedia, Colors., controls, Help & error messages design.
6. Web user interface design – jessy James Garette five layers of user experience.
7. Heuristic Evaluation: Heuristic Evaluation — Why and How?
8. visualization-, Amount of information, Focus and emphasis, Presentation information simply and meaningfully, Information retrieval on web, Statistical graphics

Text books:

1. Alan Cooper & Robert Reimann, About Face 2.0: The Essentials of Interaction Design, Wiley
2. Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale, Human-Computer Interaction (3rd Edition), Pearson, 2004.
3. Ben Shneiderman and Catherine Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition), 5th ed., Pearson Addison-Wesley, 2009
4. Donald A. Norman, The Design of Everyday Things, Basic Books, 2002

PL 2: Human Computer Interface and Agile Software Development

Group A: Human Computer Interface

Suggested list of practical:

1. Paper Prototyping using templates
2. Conducting survey interview and summarizing the result
3. Persona- conducting contextual interview and developing persona
4. GUI design- form design, menu design, help, error messages
5. Web UI design- pages, navigation, controls, (Ajax)
6. Report designs

7. Visualization and info graphics
8. Heuristic evaluation
9. Story boarding

CS 4: Agile Software Development

Learning Objective: To understand what is the software development and it's various stages like analysis, designing, coding, testing, documenting

Learning Outcomes: Students will be able to understand the various stages of software development such as analysis, designing, coding, testing, documenting.

Theory(100 marks) **Lectures 50**

1. Introduction to Agile Software Development: Understanding how traditional software development works and it's problems; Role of Agile practices in the world of software development and Tools used [6L]
2. Agile Project Planning And Management using Scrum: Requirement Analysis, Estimation techniques, Iteration planning, Introduction to development practices, user stories, sprint, backlog, velocity, burndown chart, story points [12L]
3. introduction to UML- class diagram, sequence diagram, use case diagram [7L]
4. XP & TDD : Test Driven Development & Pair Programming, Introduction to QA Practices:Fail Fast & Automated functional testing, Introduction to Continuous Integration [10L]
5. Coding and testing practices: Practicing TDD and pair programming as alternative to traditional documentation; Unit testing using Junit, test case, test suite, bad smells & Refactoring. [10L]
6. Introduction to Source Control tools - versioning, check-in/checkout, commit, branching, merging [5L]

Text books:

1. Agile Software Development with Scrum, Ken Schwaber, Mike Beedle, Prentice Hall
2. Refactoring- Martin Fowler
3. UML Distilled- Martin Fowler
4. Extreme Programming Explained: Embrace Change, First Edition by Kent Beck, Addison Wesley

Group B: Agile Software Development (Practicals)

Tools/technologies: IDEs: Eclipse/NetBeans/Visual Studio; Java, C++, C#, Ruby on Rails;

1. TDD and Refactoring exercises(Eclipse Refactoring, Junit)- 50%
    a. Using unit testing frameworks such as Junit.
    b. Creating and running tests
    c. Basic refactoring using the IDE

d. Creating test suite

e. Using debugger

2. Source and version control: (Git and Git hub; subversion)- 10%

   a. Creating repository, joining and sharing project groups

   b. Connecting to repository though IDE such as Eclipse

   c. Check-in, check-out, sharing project

   d. Creating versions, branches

3. Java Doc- 5%

   a. Generating java doc for class, method, variables, constructors

4. Build management & Dependency tools –(Maven/Ant build)- 10%

   a. Building project with dependencies

5. Defect tracking- 10%

   a. Maintaining a defect database

6. UI design and testing(desktop/web- selenium)- 10%

   a. capture/replay tools

7. Scrum- creating scrum sprint chart, burn down chart (paper based or tool based)

Semester VI

CS 5: Network Security

Learning Objective: To make the students learn the principles and practices of Cryptography, Network Security and to enable the students understand the various methods of encryption and authentication

Learning Outcomes: The student will be able to illustrate basic encryption techniques, explain security principles, asymmetric ciphers and describe digital certificates and digital signatures.

Theory(100 marks) **Lectures 50**

1. Concepts of Security & Classical Encryption Techniques [8L]

Introduction, The need for security, Security Approaches, Security Attacks, Security Services, Security Mechanisms, A Model for Network Security. Symmetric Cipher Models – Substitution techniques, Transposition techniques, Steganography, Block Cipher Operation, Electronic Code Book, Cipher Block Chaining, Block Cipher Principles, The Data Encryption Standard

2. Public Key Cryptography and Cryptographic Hash Functions [9L]

Introduction To Number Theory, Modular Arithmetic, Prime Numbers, Euler's Totient Function, Principles of Public Key Cryptosystems, The RSA Algorithm, Other Public key cryptosystems, Diffie Hellman Key Exchange.

Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Hash Functions Based on Cipher Block Chaining, MD5 Message Digest Algorithm, Secure Hash Algorithm SHA 512.

3. Message Authentication Codes and Digital Signatures [9L]

Message Authentication Requirements – Message Authentication Functions –Requirements for Security of MACs, MACs Based on Hash Functions, HMAC, MACs Based on Block Ciphers, Data Authentication Algorithm.

Digital Signatures, Digital Signature Standard.

4. Key Management & Distribution And User Authentication [9L]

Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, Public Key Infrastructure.

Text books:

Main Reading

1. Stallings William, " Cryptography and Network Security: Principles and Practises", 5th edition, Prentice Hall
2. Kahate Atul, "Cryptography and Network Security" Tata McGraw-Hill.

Supplementary Reading

1. Menezes A. J., P.C. Van Oorschot and S.A. Vanstone, "Handbook of Applied Cryptrography"

PL 3: Network Security & Multimedia Techniques

Group A: Network Security

List of suggested Assignments:

1. Implement an iterative TCP client and server application to transfer file.
2. Implement a concurrent TCP client and server application to transfer file.
3. Implement UDP client and server application to reverse the given input sentence.
4. Using basic networking command like ping, traceroute, nslookup, netstat and ipconfig (linux platform). Understanding How basic network enumeration is performed. Using who is command
5. Using Network protocol analyzer tool like ethereal or tcpdump to analyze network traffic
6. Securing(hardning) a Linux or a Windows system
7. Installing and configuring a Firewall.
8. Installing and configuring Intrusion detection system.
9. Cracking passwords and password management
10. Securing network services. i.e. Apache HTTP server.

CS 6: Multimedia Techniques

Learning Objective: To make aware the students about image, video & audio, colors models used and compression of data.

Learning Outcomes: By the end of this paper, the student will be able list the guidelines for effective creation of multimedia content., understand different types of multimedia, apply multimedia guidelines for ethical multimedia content creation, analyse different multimedia formats, compare different multimedia compression storage formats and create ethical multimedia content

Theory(100 marks) **Lectures 50**

1. Introduction to Multimedia, Social & Ethical considerations, Digital Representations, Standards [3L]
2. Introduction to Computer Graphics: Vector graphics fundamentals, shapes, transforms and filters, Bitmapped graphics: resolution, image compression, manipulation, Geometrical transformations [8L]
3. Color Science and Color Models: Human vision, Camera systems, Gamma correction, Color matching, different Color models – RGB, CYMK, Transformations among color model [8L]
4. Video: Digitising video, streamed video, video standards, compression: mpeg, dv, codec comparison, introduction to Animation: captured, sprite, key frame, web, 3-D. Virtual reality: vrml [15L]
5. Sound: Digitising and processing sound, compression, formats, MIDI. [10L]
6. Text and Layout: Text in graphics, character set, fonts, layout [6L]

Text books:

1. Nigel Chapman, Jenny Chapman; Digital Multimedia; Wiley India Edition, 2nd Edition

Reference books:

1. Ze-Nian Li & Mark S Drew; Fundamentals of Multimedia; Pearson Education International Edition

PL 3 : Group B: Multimedia Techniques

Practical can be done using Alice, GIMP, Blender, Windows movie maker, sound recorder
Suggested list of practical:

1. Design a Brochure for a given product and details, learn about different Image file formats.
2. Design a poster with given information and learn about Image compression.
3. Learn to prepare images for Print, Web and Video.
4. Edit the sound file and learn about Effects and Filters of sound.
5. Record your voice and learn about Audio Compression.
6. Record an Audio Program and Learn about streaming an audio content.
7. Learn about video editing – Prepare video with rough cut.
8. Prepare video content with title and special effects.
9. Record video content and learn about video compressions.
10. Prepare Video content for streaming.
11. Prepare an interactive presentation using flash.

CS 7: Server Side Web development

Learning Objective: To give an understanding the web software development: how it is different, issues involved in it.

Learning Outcomes: The students will be able to:

- Explain Web Technology and the difference between standalone application and the Web technology.
- Develop web applications to solve various real life problems using the server side web scripting language – PHP.
- Maintain security of the web based systems against the potential intruders.
- Design interactive systems using the concept of AJAX.

Theory(100 marks) **Lectures 50**

1. Web Technologies: Introduction to Web technology, Web pages and Browsing, Dynamic Web Pages, Java script, Dynamic web document technologies - PHP, JSP, ASP, Active web pages and Active Web technologies.
2. Tags, Escaping from HTML, Types: Resources, NULL, Callbacks, Type juggling.
3. Variables: Basics, Predefined variables and Scope, Constants: Syntax, Magic constants, Expressions.
4. Operators, Control structures, Functions, Predefined exceptions
5. Security: Introduction, General considerations, Installed as CGI binary, Installed as an Apache module, Filesystem Security, Database Security, Error Reporting, Using Register Globals, User Submitted Data, Hiding PHP
6. Features: HTTP authentication with PHP, Cookies, Sessions, Handling file uploads, Connection handling, Persistent Database Connections, DTrace Dynamic Tracing
7. Ajax - request object creation, forwarding the request, accepting response object and display on webpage

Web Reference:

1. http://in1.php.net/manual/en/index.php for PHP v 5.5 and above

PL 4: Server Side Web Development & Embedded Systems

Group A: Server Side Web Development

Following is a suggested list of practicals:

1. Install and configure Apache, PHP and MySQL
2. Authentication, Cookies, Session management
3. Error handling
4. Database connectivity
5. Using framework like CodeIgniter
6. Ajax implementation

7. File uploading
8. Uploading and images to and from server

CS 8: Embedded Systems

Learning Objective: To understand how to model the real world in computers and design & test the solution before implementing.

Learning Outcomes: Students will be able to understand how to model the real world in computers and design & test the solution before implementing.

Theory(100 marks) **Lectures 50**

1. **Introduction to embedded system:**

A First Look at Embedded systems- Examples of Embedded systems-applications area-categories of embedded system – recent trends in embedded system:

2. **Design challenge** –

optimizing design metrics Common design metrics- Processor technology- General-purpose processors – software - Single-purpose processors – hardware- Application-specific processor-IC technology

3. **Architecture of embedded system:**

hardware architecture –software architecture - Programming for embedded system.

4. **The process of embedded system development**

Interrupts-Microprocessor-Architecture-Interrupt Basics-The Shared-Data Problem-Interrupt Latency-shared data problems- survey of software architecture -Round-Robin-Round-Robin with Interrupts-Interrupt Latency –RTOS.

5. **RTOS**

architecture of kernel –task and task scheduler-interrupt service routines- semaphores-message queues- mail boxes-pipes –events-timer – memory management –interrupt routines in RTOS Environment-overview of embedded /real time operating system .

6. **Embedded Software Development Tools**

-Host and Target Machines-Linker/Locators for Embedded Software-Getting Embedded Software into the Target System.

7. **Debugging Techniques-**

Testing on Your Host Machine-Instruction Set Simulators-The assert Macro-Using Laboratory Tools

8. **An Example System-**

What the Program Does-Environment in which the Program Operates

9. **Task Image creation** –

operating system software –target image creation for windows XP embedded-porting RTOS on a micro-controller based development board.

10. **Representative embedded systems-**

programming in Linux-programming in RTLinux-Development of Navigation System – Development of protocol converter-mobile Java applications.

11. **State machine and concurrent process models** –

Introduction- Models vs. languages, text vs. graphics- Models vs. languages- Textual languages versus graphical languages-An introductory example- A basic state machine model: finite-state machines (FSM)- Finite-state machines with data path model: FSMD- Using state machines-

Describing a system as a state machine-Comparing the state machine and sequential program model- Capturing a state machine model in a sequential programming language- Hierarchical/Concurrent state machine model (HCFSM) and the State charts language

12. **Program-state machine model (PSM) –**

The role of an appropriate model and language

13. **Concurrent process model-**

Use of Esteral language for embedded software development.

Text books:

1. Embedded software primer by David Simon – Pearson

Reference books:

1. Art of embedded system by Jack Ganssle
2. Embedded systems Architecture by Tammy Noergaard – Elsevier publications
3. Embedded /Real time systems – by DR.K.V.K.K.Prasad. – Dreamtech
4. Esterel language by Gerard Berry (web site reference)
5. Embedded system design by Arnold S.Berger
6. model checking by Edmund M.clark
7. Embedded Systems Building Blocks by Jean LaBrosse
8. Embedded Systems Design by Arnold Berger

The Art of Programming Embedded Systems by Jack Ganssle

Semester III

Foundation Course : Introduction to Cyber security and cyber law

Learning Objective: To give an overview of the computer, networking and why and how of securing information in e-World.

Learning Outcomes:

The students will be able to:

- Explain the concept and usage of Operating Systems, their directory structures, software installation etc.
- Describe the basics of computer networks.
- State the various cyber-crimes taking place in the cyber world.
- Explain the cyber laws that have been framed in order curb the related crimes.
- Explain the concept of cyber forensics and ways to recover lost data that is lost either due to some unintentional or intentional neglect.

Theory(100 marks) Lectures 60

1. Operating Systems [5L]

   Windows OS, Linux OS: Directory structure, Environment variables setting and use, Users rights and security, Services, Software installation.

2. Basics of Computer Networking [10L]

Networking basics, why networking of computer is needed, Network media, topology, Types, Internet: role and importance, IP Addressing - public Vs Private, Static Vs Dynamic; WWW & related protocols, Intranet & Extranet, Internet Infrastructure, Search Engines, Web Browser, Web server.

3. Cyber Crimes [5L]

Introduction, types of cyber crimes, its impact: social, personal, financial; reporting cyber crimes

4. e-World security [10L]

Threats in Cyber Space, Classification of threats, BYOD and portable devices threats, 0-day attacks, insider threats, Cyber Warfare, Malware threats, mobile apps threats Social media and its safe uses: Social media- its usages, Social networking- types, uses, importance, social networking safety

5. Online Privacy [10L]

Privacy : basic concepts, Sensitive personal information, Privacy policies, Google/Facebook privacy policies, Privacy laws, IPR, Ethics & safe practices

6. Cyber Laws [15]

Evolution and purpose, offence & defence, bailable and non-bailable offences, provisions related to e-commerce, provisions for cyber crimes, adjudicating officers, CERT-IN- its role and powers

7. Cyber Forensic [5L]

Data recovery, evidence collection, cloning of devices, media sanitization

Web Resources:
www.linux.org ,www.cert.org ,www.sans.org ,www.us.cert.gov , en.wikipedia.org/wiki/List_of_cyber_attack_threat_trends ,compnetworking.about.com/od/basicnetworkingconcepts

Semester IV

Foundation Course : Information Systems & IT Entrepreneurship

Learning Objective: To study the different aspects of human computer interaction and the computer interface design concepts.

Learning Outcomes:
The students will be able to:

- Explain various information systems in use in the market.
- Explain the working of e-commerce in our day-to-day lives.
- Generate new ideas with regards to business start-ups with the help of IT.
- Discuss the workflow of Enterprise Resource Planning packages.
- Explain the current and future trends in computing.

Theory(100 marks) Lectures 60

1. Information Systems & its role in business- operational, management, strategic. Data & Information. Types of information systems- TPS, MIS, DSS, EIS. – [3L]
2. e-Commerce- Concept and Models : Introduction to E-Commerce -Definition and scope of E-Commerce and M-Commerce, E-Commerce trade cycle, Electronic Markets, Internet Commerce, Benefits and Impacts of E-Commerce. Elements of E-Commerce –varoius elemennts, e-visibility, e-shops, Delivery of goods and services, Online payments, After-sales services, Internet E-Commerce security . EDI and Electronic Payment Systems Electronic Payment Systems: credit/debit/smart cards, e-credit accounts, e-money , EDI. Introduction to EC models -Inter-organization and intra-organization E-Commerce, E-Commerce Models: B2B, B2C, C2B, C2C, G2C, C2G. Practices in E-Commerce . E-Business-Introduction to Internet bookshops, Grocery Suppliers, Software Supplies and support, Electronic newspapers, Virtual auctions, Online share dealing, e-diversity . Mobile Commerce and Future of E-Commerce -Introduction to Mobile Commerce, Benefits of Mobile Commerce, Impediments of M-Commerce, M-Commerce framework, Emerging and future trends – [15L]
3. ERP- Enterprise-An Overview-. Introduction to ERP, Basic Concepts of ERP, Risks and Benefits of ERP, ERP and Related Technologies. ERP Marketplace and Marketplace Dynamics. ERP-Functional Modules. ERP Implementation -ERP Implementation Basics, ERP Implementation Life Cycle, ERP Package Selection, ERP (Implementation) Transition Strategies, ERP Implementation Process, ERP Project Teams, Consultants, Vendors, and Employees, Success & Failure Factors of an ERP Implementation. ERP Operation & Maintenance, Maximizing the ERP System- [12L]
4. SCM, CRM, Knowledge management, Business Intelligence- [6L]
5. Information systems development & maintenance approaches, IT outsourcing- [9L]
6. current & future trends- cloud computing, mobile & ubiquitous computing, big data and its impact on information systems- [3L]
7. Entrepreneurship & Start-ups- business models, venture capital etc. success & failure stories- lean start-up model, innovation [12L]

Text Book:

1. Enterprise Resource Planning, ALEXIS LEON, ISBN: 9780070656802,2007
2. Kenneth C. Laudon, Jane P. Laudon; Management Information Systems; Prentice Hall; 12th Edition
3. The lean Startup- Eric Ries, Penguin India