ANNEXURE II

Parvatibai Chowgule College of Arts and Science (Autonomous)

DEPARTMENT OF COMPUTER SCIENCE

COURSE STRUCTURE

THREE YEAR B.Sc. DEGREE COURSE IN COMPUTER SCIENCE (Offered to Students who had taken admission to F.Y.B.Sc in the year 2019-20)

SEMESTER	CORE COMPULSORY		CORE ELECTIVE			
I	**COM-I.C-1** Mathematical foundation of Computer Science – I	**COM-I. C-2 *** Introduction to Programming	---	---	---	---
II	**COM-II. C-3A **** Database Management System I	**COM-II.C-4 *** Data Structures	---	---	---	---

III	**COM-III.C-5A *** Object Oriented Programming	---	**COM -E1** Software Engineering	**COM-E2** Digital Logic Design	**COM-E3** Mathematical Foundation of Computer Science – II	**COM-E4** Web Designing
IV	**COM-IV.C-6** Computer Architecture and Organization	---	**COM - E5** Design & Analysis of Algorithms	**COM-E10** Mobile Application Development	**COM-E7** Server Side Programming	**COM -E8** **HCI**
V	**COM-V.C-7 *** Operating Systems	---	**COM-E9** Embedded Systems	**COM-E17** Data Base Management System II	**COM-E11** Introduction to Data Science	**COM-E12** Software Testing

VI	**COM-VI.C-8 *** Computer Networks	---	**COM- E-13** Network Security	**COM- E-14** Cloud Computing	**COM-. E-15** Multimedia Techniques	**COM-VI. E-16** Digital Marketing

Note: * Core Compulsory Courses also offered for minor subject combination.

** Core Compulsory Courses also offered for minor subject combination in 4th Semester.

Programme Specific Outcome (PSO) for B.Sc. Computer Science

At the end of the programme the students will be able to:

PSO1: Analyze a Software problem, design, implement a solution and evaluate the proposed solution to ensure that it meets customer needs and Software standard.

PSO2: Use and Apply appropriate current technologies, techniques and modern tools necessary for computing practice.

PSO3: Embark on an Entrepreneurial venture or be eligible for employment in IT industry or pursue higher education.

PSO4: Apply the concept of networking and security features in designing the systems.

Parvatibai Chowgule College of Arts and Science (Autonomous)
Margao, Goa

B.Sc. Computer Science – Syllabus
Offered from (2020-21)

Semester I

Course Title: Mathematical Foundation of Computer Science I
Course Code: COM-I.C-1
Marks: 75
Credits: 3
Duration: 45 Hrs

Prerequisite Courses: Nil

Course Objectives:

- To build mathematical foundations that are essential requirement in understanding various concepts related to computer science.

Course Outcome:
At the end of the course students will be able to:

CO1: Explain various fundamental concepts.
CO2: Convert a given number from one base to another.
CO3: Apply counting principles to determine probabilities.
CO4: Demonstrate an understanding of relations and functions and determine their properties.
CO5: Evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.
CO6: Write an argument using logical notation and determine if the argument is valid or not.
CO7: Construct and analyze finite state automata.

SYLLABUS

UNIT I: Combinatory and Number systems [15 HRS]
Permutations; Combinations; Counting; Summation; generating functions; recurrence relations.
Binary Number System-Decimal to binary conversion and vice versa, binary number representation (signed, 1‘s Complement and 2‘s complement) binary addition, subtraction, binary to octal, hexadecimal conversion and vice versa. Floating point representation.

UNIT II: Boolean Algebra, Set, Relations and Functions [15HRS]

Boolean functions, truth table, De Morgan's theorem, logic gates, Realization of Boolean Function using logic gates, Simplification using Karnaugh map.

Set-Venn diagram, set operations, relations and properties, closures, equivalence, relations, ordering

Functions-function types, inverse of functions, composition of Partial functions, recursive functions, growth of functions.

UNIT III: Logic, Grammars, Languages and Automation [15 HRS]

Propositional logic, first order logic, mathematical induction, deduction, proof by contradiction, program correctness. Grammars and languages, finite automation of finite state machines, regular languages, regular expressions.

REFERENCES:

MANDATORY:

1. Rosen, K. H., & Krithivasan, K. (2012). *Discrete mathematics and its applications: with combinatorics and graph theory*. Tata McGraw-Hill Education.

SUPPLEMENTARY:

1. Sarkar, S. K. (2016). *A Textbook of Discrete Mathematics*. S. Chand Publishing.

WED BASED:

1. https://web.stanford.edu
2.https://www.cse.iitb.ac.in

Practical: Mathematical Foundation of Computer Science I
Credit: 1
Marks: 25
Duration: 30 Hrs

Programs to be written using C Language:

1. Generate all permutations of n symbols, where 2 <=n<= 5 is user defined.

2. Read a number and convert to other number formats.

a) Convert an integer into binary number
b) Convert a binary number to decimal number.
c) Convert a binary number to octal number.
d)Convert a number into normalized form.

3. String Manipulation

a) Read a string of decimal digits. Find the frequency distribution of digits.
b) Read a binary string. Check the occurrence of the pattern 1001 in the string.

c) Read two binary numbers. Add them using 1‘s complement and 2‘s complement method.

4. Read two integer numbers. Find their GCD using recursion.

5. Read the value of p. Find the p-th Fibonacci number from the following recurrence relation.
$f(0) = 0$, $f(1) = 1$, $f(n) = f(n-1) + f(n-2)$, $n \geq 2$.

6. Given two functions $f(x) = x^3 + 2x + 3$, and $g(x) = 3x^2/4 + 10$,
find $f \circ g(x)$.

7. Read an expression containing parentheses and check whether it is properly parenthesized.

a) Equal number of (and)‘ brackets
b) Equal number of { and }‘ brackets
c) Equal number of [and]‘ brackets s

8. Applications of set theory
a) Read a set and check whether a given number is a member of the set.
b) Read two sets. Find their union.
c) Read two sets. Find their intersection.

9. Applications of finite state machines, matrices, Boolean algebra, gates.

10. Bit-wise operations using C

****

Course Title: Introduction to Programming
Course Code: COM-I.C-2
Marks: 75
Credits: 3
Duration: 45 Hrs

Pr-requisites courses: Nil

Course Objectives :

- To understand the concept of basic computer algorithm and flowchart and use the algorithm for various problem solving.
- To implement algorithms using high level programming language.
- To understand basic principles of structured programming – example C.

Course Outcome: Upon completion of the course students will be able to:

CO1: Explain problem solving strategies.
CO2: Draw a flowchart for a given problem.
CO3: Write an algorithm for a given problem.
CO4: Explain and Apply sorting and searching algorithms.
CO5: Recognize and incorporate programming elements such as loops, decision making, functions, arrays, string.
CO6: Recognize and incorporate programming elements such as structures, pointers and files into applications that solve real world problems.

SYLLABUS

UNIT I **[10 HRS]**

Introduction to Computer Problem Solving : Algorithm, Flowchart, The Problem-Solving Aspect, General problem-solving strategies, Top-Down Design, Implementation of Algorithms, Efficiency of Algorithms, Recursive algorithms.

Basic Algorithms : Exchanging the values, Summation of a set of numbers, factorial computation, generation of the Fibonacci series, reversing the digits of an integer, base conversion.

Factoring Methods : Finding divisors of an integer, finding the Greatest Common Divisor of two integers, generating prime numbers, computing prime factors of an integers.

Sorting and Searching : Bubble sort, Insertion Sort, Sequential Search and Binary Search.

UNIT II **[20 HRS]**

C Language : History, Structure of a C program, Keywords, Identifiers, variables, constants, data types, Arithmetic Operators & Expressions, Logical operators and Relational Operators, Precedence and Associativity rules.

Conditions and Iterations : Conditions and Actions, Condition statement, Simple control statement (if, if-else,switch), Iterative control statements (for, while, do-while).
Functions : What is a function, Advantages of functions, Standard library functions; User define functions – declaration, definition, function call, parameter passing, return keyword. Scope of variables, Storage classes, Recursion.

Arrays : One- and Two-dimensional arrays: Array declaration, initialization, accessing the values, passing arrays to functions.

Pointers : Pointer declaration, initialization, Pointer arithmetic, Pointer to Pointer, Arrays and Pointers, Functions and Pointers – passing pointers to functions, function returning pointer, dynamic memory allocation.

UNIT III **[15 HRS]**
Strings : Declaration and initialization, standard library string functions, strings and pointers, array of strings.

Structure and Union : Creating structures, accessing structure members, array of structures, passing structure to functions, nested structure, pointers and structures, union, difference between structures and unions.

File Handling : FILE variable, file access modes, operations on files, random access to files, command line arguments.

Pre-processing : Format of Preprocessor directive, File Inclusion directive, Macro substitution, conditional compilation.

REFERENCES:

MANDATORY:

1. Dromey, R. G. (1982). *How to Solve it by Computer*. Prentice-Hall, Inc..
2. Kanetkar,Y. (2012). *Let us C,* BPB Publications,
3. Forouzan, B. A., & Gilberg, R. F. (2000). *Computer Science: A structured programming approach using C*. Brooks/Cole Publishing Company..

SUPPLEMENTARY:

1. Horowith,E.,Sahni ,S.,Sanguthevar,R. (2008). *Fundamentals of Computer algorithm*, Orient Longman.
2. Gottfried,B. (2010). *Programming with C*, Tata McGraw Hill.

WED BASED:

1. GNU GCC (GNU Compiler Collection) @http://gcc.gnu.org, with source codes.
2. Bjarne Stroustrup's C++11 FAQ @http://www.stroustrup.com/C++11FAQ.html.
3. https://www.tutorialspoint.com/cprogramming
4. https://www.javatpoint.com/c-programming-language-tutorial
5. https://www.w3schools.in/c-tutorial/
6. https://www.guru99.com/c-programming-tutorial.html
7. https://www.geeksforgeeks.org/c-programming-language/
8. E Book -https://www.edutechlearners.com/download/books/Let%20Us%20C%20by%20Yashavant%20Kanetkar%20PDF.pdf
9. E Book -http://www2.cs.uregina.ca/~hilder/cs833/Other%20Reference%20Materials/The%20C%20Programming%20Language.pdf
10. E Book - http://www.kciti.edu/wp-content/uploads/2017/07/cprogramming_tutorial.pdf

Practical: Introduction to Programming
Credit: 1
Marks: 25
Duration: 30 Hrs

Programs using C language that covers the following concepts:

1. Conditions
   - if..else
   - nested if

2. Iterative Control Statements
   - for
   - while
   - do…while

3. Functions.
   - Standard Library functions
   - Call by Value
   - Call by reference
   - Recursive functions

4. Arrays.
   - One Dimensional Arrays
   - Two Dimensional Arrays

5. Sorting
   - Bubble sort
   - Insertion sort

6. Searching.
   - Sequential search
   - Binary search

7. Pointers.
   - Arrays and Pointers
   - Function returning pointers
   - Dynamic memory allocation

8. Strings.
   - Standard Library string functions
   - Strings and Pointers
   - Array of Strings

9. Structure and Union

- Array of structures
- Passing Structure to functions
- Nested structure
- Structure and Pointer
- Union

10. File Handling.
    - Text file
    - Binary file
    - Random Access to a file
    - Command Line arguments

****

Semester II

Course Title: Data Base Management Systems -I
Course Code: COM-II.C-3A
Marks: 75
Credits: 3
Duration:45 Hrs

Prerequisites: Nil

Course Objectives:

- To provide basic knowledge of a database management system.
- To understand importance of Entity Relationship diagram.
- To formulate queries in Relational Algebra and SQL for Database manipulation.
- Familiarity with any RDBMS during practical sessions.

Course outcomes:

At the end of the course students will be able to:

CO1: Explain the various database concepts and the need for database systems.
CO2: Identify different entities and relationship between them.
CO3: Represent the given system using ER diagram.
CO4: Convert an ER diagram to a database schema.
CO5: Formulate queries in Relational Algebra, SQL to manipulate the database.
CO6: Analyze the schema to see if they fulfill Normalization criterion.
CO7: Design database using appropriate RDBMS.
CO8: Design User Interface for Database.
CO9: Design Reports for Database.

SYLLABUS:

UNIT I Overview of DBMS, Design and ER model. [15HRS]
Overview of database management : Data, information, database, database management system; Managing data; File systems versus a DBMS, advantages of a DBMS; Data abstraction, instances and schemas, data models; Data manipulation language, data definition language; Architecture of a DBMS; Users of a DBMS, database administrator.

Database design and the ER model : Design phases – conceptual design, logical design, physical design; ER model – entities, attributes, and relationships, mapping cardinalities, keys; ER diagrams – strong entities, weak entities, generalization, specialization, aggregation; Converting ER diagram to relational schemes.

Unit II : Functional dependency and normalization [10HRS]
Atomic domain, nested relation; Key, super key, primary key, candidate key; Functional dependency, axioms, closure of a set of attributes, closure of a set of functional dependencies; Purpose of normalization; 1NF; 2NF;

3NF; BCNF.

UNIT III : Relational model and SQL [20HRS]

Relational model :Relation, properties of relational model; Entities, integrity constraints, referential integrity constraints; Relational algebra – select, project, cross product, set operations, rename operation; Other relational operations – natural join, outer join.

SQL: Basic structure of SQL query – Create, select, where, from, rename operation; Set operations; Aggregate functions; Group by, having clauses; DDL Nested queries; Views; Insert, delete, update.

REFERENCES:

MANDATORY:

1. Silberschatz, A. (2006). Database system concepts.—6th ed.

SUPPLEMENTARY:

1. Ramakrishnan, R., & Gehrke, J. (2000). *Database management systems*. McGraw Hill.
2. Elmasri, R., & Navathe, S. (2017). *Fundamentals of database systems*. Pearson.

WEB BASED:

1.https://www.db-book.com/db6/slide-dir/ (Sixth edition Powerpoint , PDF of A Silberschatz, H F Korth, S Sudarshan, Database system concepts,)
2.https://www.db-book.com/db7/
3.https://www.tutorialspoint.com › dbms
4.https://www.w3schools.in › dbms
5.https://www.studytonight.com › dbms
6.https://www.oracletutorial.com

Practical: :Database Management Systems I
Credit :1
Marks :25
Duration: 30 Hrs

1. ER diagram (1P)
2. ER diagram with specialization/generalization and aggregation.(1P)
3. Converting ERD into Schema.(2P)
4 SQL (2P)
5 Nested Queries (2P)
5Normalization(2P)
6 Report Writing (1P)
7. Mini project (4P)

****

Course Title: Data Structures
Course Code: COM-II.C-4
Marks: 75
Credits: 3
Duration:45 Hrs

Prerequisite Courses: Nil

Course Objectives:

- To impart the basic concepts of data structures and algorithms.
- To understand concepts about searching and sorting techniques.
- To Understand basic concepts about stacks,queues,lists,trees and graphs.
- To understanding about writing algorithms and step by step approach in solving problems with the help of fundamental data structures.

Course outcomes:

At the end of the course students will be able to :

CO1 : Define relevant standard algorithms for various data structures. Learn various applications of data structures.
CO2 : Implementation of data structures.
CO3 : Use various data structures for sorting and searching.
CO4 : Analyze and compare algorithms for efficiency using Big-O notation.
CO5 : Formulate new solutions for programming problems.

SYLLABUS

UNIT I: **[15HRS]**

Introduction to data structures:
Concept, Data type, Data object, ADT, Need of Data Structure, Types of Data Structure.

Algorithm analysis:
Algorithm – definition, characteristics, Space complexity, time complexity, Asymptotic notation (Big O).

Linked List:
Introduction to List, Implementation of List – static & dynamic representation, Types of Linked List, Operations on List, Applications of Linked List, polynomial manipulation, Generalized linked list – concept & representation.

Stacks:
Introduction, Representation-static & dynamic, Operations

UNIT II **[15HRS]**

Stack Applications :

Application - infix to postfix& prefix, postfix evaluation, Simulating recursion using stack.

Queues:

Introduction, Representation -static & dynamic, Operations, Circular queue, priority queue (with implementation), Concept of doubly ended queue.

Trees:

Concept & Terminologies, Binary tree, binary search tree, Representation – static & dynamic, Operations on BST – create, Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes, non recursive in order traversal, Expression Tree. Introduction to AVL Trees

UNIT III **[15HRS]**

M-Way Search Trees : Introduction, B Tree, B+ Tree.

Searching and Sorting :

Use of various data structures for searching and sorting, selection sort, merge sort, quick sort, heap sort and hashing.

Graph :

Concept & terminologies, Graph Representation – Adjacency matrix, adjacency list, Traversals – BFS & DFS, Application of BFS, DFS – Shortest path, Backtracking.

REFERENCES:

MANDATORY:

1.Horowitz,E., Sahni,S. (2008). *Fundamentals of Data Structures in C*, University Press.

SUPPLEMENTARY:

1. Langsam Yedidyah, Augenstein J. Moshe, Tenenbaum M. A aron ,(2018),*Data Structures using C*, Pearson Education

2. Richard.G, Behrouz.F, *Data Structures: A Pseudocode Approch with C*, Cengage Learning.

WEB BASED:

1. https://www.tutorialspoint.com/data_structures_algorithms
2. https://www.w3schools.in/data-structures-tutorial
3. https://www.studytonight.com/data-structures/
4. https://www.programiz.com/dsa
5. https://www.geeksforgeeks.org/data-structures/
6. https://www.javatpoint.com/data-structure-tutorial
7. E Book - https://www.scribd.com/doc/261233741/Data-Structures-Through-C-Yashavant-Kanetkar

Practical: Data Structures
Credit: 1
Marks: 25
Duration: 30 Hrs

Programs using C language that covers the following concepts:

1. Stack: Static/Dynamic stack implementation.
2. Stack: infix to postfix.
   Stack: Evaluation of Postfix expression.
3. Queues: Static and Dynamic Queue Implementation
   Queues: Circular queue
4. List: Singly Linked List,
5. List: Doubly Linked List
6. List: Circular Linked List
7. Linked List: Polynomial addition
8. Trees: Binary Search Tree: create, add, delete, display nodes.
9. Trees: BST traversal.
10. Graph: Representation of Graphs, Graph Traversals.
    Graph: DFS, BFS.

****

SEMESTER III

Course Title: Object Oriented Programming
Course Code: COM-III.C-5A
Marks : 75
Credits: 3
Duration:45 Hrs

Prerequisite Courses : Nil

Course Objectives:

- To teach the basic concepts and techniques which form the object oriented programming paradigm.
- To introduce object oriented programming (OOP) using Java.

Course Outcomes:

At the end of the course students will be able to :

CO1 : Apply fundamental object-oriented concepts in problem solving.
CO2: Analyze problem scenario and identify classes/objects, their properties/functionalities and associations.
CO3 : Analyze the problem scenario and model the system using UML diagrams.
CO4 : Implement the object oriented model in any object oriented language.

SYLLABUS:

UNIT I: Introduction **[15 HRS]**
Principles of OOP
Programming paradigms.Basic concepts in OOP. OOP: major principles - encapsulation, abstraction,inheritance, polymorphism. Benefits of OOP. Applications of OOP.
Introduction to Java
Basics of Java programming, Data types, Variables, Operators, Control structures including selection,Looping, Java methods, Overloading, java.Math class, Arrays in java.

UNIT II Principles of OOP **[15 HRS]**
Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects,object serialization, Inbuilt classes like String, Character, StringBuffer, File, this reference.
Inheritance and Polymorphism
Inheritance in java, Super and sub class, Overriding, java.lang.Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java,
Package in java, java.util package.

UNIT III: Exceptions, Multi-threading and GUI programming: **[15 HRS]**
Event and GUI programming
Design patterns – what and why? It's classification. Introduce the Observer design pattern.
Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers:
Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons,Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box,Applet and its life cycle.
Multi-threading in java
Multi-threading in java, Thread life cycle and methods, Runnable interface, Thread synchronization.
Exception handling
Exception handling – what and why? Try and catch block. Multiple catch blocks. Nested try, finally block,throw keyword, throws keyword. Custom Exception.Introduction to the Collections Framework.

REFERENCES:

MANDATORY:

1. Deitel, P., & Deitel, H. (2011). *Java How to program*. Prentice Hall Press.
Supplementary:
2. Naughton.P, Schildt.H, (2000), *Java 2 – The Complete Reference* TMH publications
3. Patrick,N. (1997). *The Java Handbook* –TMH publications
4. Mughal, K. A., & Rasmussen, R. W. (2003). *A programmer's guide to Java certification: a comprehensive primer*. Addison-Wesley Professional.
5. Flanagan, D. (2004). Java examples in a nutshell.
6. Arnold, G., H.(2005)"*The Java Programming Language"* Addison-WesleyProfessional,

WEB BASED:

1.www.javapoint.com
2.www.tutorialspoint.com
3.www.gurugg.com
docs.logout.org>Programmation>Java>Programming with Java_A primer

Practicals: Object Oriented Programming
Credit: 1
Marks: 25
Duration: 30 Hrs

Programs using Java language that covers the following concepts:

1) Classes and instances
2) Working with the java.Math class
3) Inheritance
4) Composition v/s inheritance
5) Polymorphism, abstract classes and interfaces
6) Algorithm and Data Structures
7) Utilizing the java.util package
8) Event handling and GUI
9) Applets
10) I/O programming
11) Exception handling
12) Multi-threading
13) Collections framework

****

Course Title: Software Engineering
Course Code: COM-E1
Marks: 75

Credits: 3
Duration: 45 Hrs

Prerequisite Courses : Nil

Course Objectives:

- To Understand the various software development methodologies and estimation of
- software projects
- To analyze and design software projects
- To study the various phases of a S/W Development Project.

Course Outcomes:

At the end of the course students will be able to:

CO1: Understand the various Software Development Methodologies
CO2: Apply Estimation techniques to live projects
CO3: Analyze Software Projects.
CO4: Design Software Projects.

SYLLABUS:

UNIT I: **[15 HRS]**
SOFTWARE PROCESS:
Characteristics of software process.
Software Development Processes and Methodologies: waterfall, prototyping, iterative, spiral,unified process.Benefits of iterative and incremental approach with emphasis on Unified process, CASE Tools,Agile methodologies.

PROJECT MANAGEMENT:
Planning a Software Project – Cost estimation, Project Scheduling, Software configuration management plans, Quality Assurance plans, Project Monitoring plans and Risk Management.Techniques such as Interviewing , Requirement Workshop, brainstorming, prototyping.Characteristics of SRS.

UNIT II: **[15 HRS]**
OOAD and UML:
OOAD: Definition; object oriented analysis; object oriented design and modeling; Assigning responsibilities.
UML: Main UML diagrams – Class diagram , sequence diagram, activity diagram, use case diagram.
Use case model – use case diagram , use case descriptions, use case realization using sequence and activity diagrams. Supplementary requirements. Advanced use case model
features.

Requirements: Functional and non-functional
System Design : Class diagram, sequence diagram, activity diagram, state chart diagram,deployment diagram. Brief introduction to other UML diagrams.

SOFTWARE ARCHITECTURE PATTERNS:
Major Architectural Styles (patterns) like Layered Architecture, Pipe and Filter, Shared (Central)Data Store, Event Driven, Model-View-Controller (MVC), "Distributed & Emerging" Service Oriented Architecture (SOA) and Elementary GRASP Patterns.

UNIT III **[15 HRS]**
HUMAN COMPUTER INTERACTION:

HCI Definition; User categories, Interface Design-Internal & External Interface design, user interface design, Interface design guidelines.
CODING:
Coding styles, standards, peer reviews, checklist.
TESTING:
Testing Fundamental, Functional Testing, Structural Testing, Testing Object-Oriented Programs,
Testing Process and Metrics.
DOCUMENTATION and MAINTENANCE:
Need for Software Documentation. Types of documentation, Need for Maintenance; Types of Maintenance.
REENGINEERING:
Business Process Reengineering, Software Reengineering, Reverse Engineering,Restructuring,
Forward Engineering, The Economics of Reengineering.

REFERENCES:
MANDATORY:

1. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
2. Larman C.,(2015) *Applying UML and patterns*. Addison Wesley.
3. Bourque, P., & Fairley, R. E. (2014). *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press.

SUPPLEMENTARY:

1. Jalote, P. (2012). *An integrated approach to software engineering*. Springer Science & Business Media.
2.Sommerville I.,(2015) Software Engineering.Adison Wesley.
3. Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Modeling Object Language. *Object Technology Series, 3rd edition, Addison-Wesley*.

WEB BASED:

1.https://www.tutorialspoint.com/software_engineering
2.https://www.w3schools.in/sdlc-tutorial
3.https://www.geeksforgeeks.org/software-engineering
4.https://www.javatpoint.com/software-engineering-tutorial

Practicals : Software Engineering
Credit : 1
Marks : 25
Duration: 30 Hrs

List of suggested PRACTICALS :
For a given project/case study
1) Requirements Gathering Techniques [2P]
2) Gantt Chart [2P]
3) USE Case diagram and Use Case descriptions for the Use Cases [3P]
4) Class Diagram [2P]
5) Sequence Diagram [2P]
6) Activity Diagram [2P]
7) State Chart Diagram [2P]

****

Course Title: Digital Logic Design
Course Code: COM - E2
Marks: 75
Credits: 3
Duration: 45 Hours

Prerequisite Courses: Nil

Course Objectives:

- To understand the basics of Boolean algebra and the operation of logic components, combinational, sequential circuits and design of digital systems.

Course Outcomes:

After completing this course, students will be able to:

CO1: Convert values between various number systems/codes.
CO2: Simplify the logical expression using Boolean algebra.
CO3: Design, simplify and implement combinational logic circuits.
CO4: Design and implement the sequential logic circuit and their applications.
CO5: Explain the methods of D/A converter and A/D converters (Successive Approximation method).
CO6: Explain the classifications and characteristics of semi conductor memories with memory expansion.

SYLLABUS:

UNIT I:Number Systems,Boolean Algebra and Convertors: [15 HRS]
Number Systems: Decimal, Binary, Hexadecimal, and Octal number systems; BCD Code, Gray Code, Excess-3 Code, ASCII code, Unicode.
Boolean Algebra: Basic Boolean functions, Postulates and theorems of Boolean Algebra, logic gates, Sum-of-Products and Product-of-Sums forms of Boolean functions;Canonical and Standard forms. Simplification of Boolean Functions, Plotting of K-Maps, POS and SOP Simplification, NAND and NOR implementation.
D/A & A/D Converters: Digital to Analog converter – Introduction, Weighted-Resistor & R-2R Ladder,Analog to Digital Converter – Introduction, Successive Approximation method.

UNIT II:Combinational and Sequential Circuits: [15 HRS]
Combinational Circuits: Design procedure for combinational logic circuits; design andanalysis of Half Adder, Full Adder; their use in designing other combinational logic circuits; Analysis & Design of Encoders and Decoders; Multiplexer and demultiplexers;
their use in designing combinational circuits.
Sequential Circuits (Registers):SR, JK, T, D Flip-flops and Latches, their schematic symbols, Truth tables. Shift Registers - SISO, SIPO, PISO, PIPO, Bi-directional Shift Registers, Loading methods for Shift, Registers.

UNIT III:Sequential Circuits(Counters) and Semiconductor memories: [15 HRS]
Counters: Design and analysis of Counters: Synchronous Counters, Modulo Counters,Asynchronous, Ripple and Ring Counters; Application of Counters.
Semiconductor memories:Memory organization and operation, expanding memory size,Classification and characteristics of memories, RAM, ROM, Synchronous DRAM SDRAM),Double Data Rate SDRAM, Synchronous SRAM, DDR and QDR SRAM,Content Addressable Memory.

REFERENCES:

MANDATORY:

1. Jain, R. P. (2003). *Modern digital electronics*. Tata McGraw-Hill Education.

SUPPLEMENTARY:

1. Leach.D, Malvino,M , Saha.G*, Digital Principles and Applications*, ,Mc. Graw Hill (SiE).
2. Mano, M. M. (2017). *Digital logic and computer design*. Pearson Education India.
3. Taub, H., & Schilling, D. L. (1977). *Digital integrated electronics*. McGraw-Hill College.

WEB BASED:

1. https://nptel.ac.in/courses/117/106/117106114/
2. https://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials/
3. https://logisim.en.uptodown.com/windows
4.https://www.edutechlearners.com/download/books/Morris%20Mano%20Digital%20Design%203rd%20Edition.pdf
5.https://books.google.co.in/books?id=dnq3HmDN1ZAC&printsec=frontcover&redir_esc=y#v=onepage&q&f=false

Practicals :Digital Logic Design
Credits :1
Marks :25
Duration: 30 Hrs

Practicals:

1. Introduction to digital ICs, specifications, verification of the truth tables of logic gates1 (2 Lab sessions)
2. Introduction to digital ICs, specifications, verification of the truth tables of logic gates2. (2 Lab sessions)
3. Implementation of the given Boolean function using logic gates in SOP form . (adder/subtractor -2 sessions)
4. Decoder and Encoder (2 sessions).
5. Multiplexer –Demultiplexer (2 sessions).
6. Verification of state tables of RS, JK, T and D flip-flops.
7. Design and verification of the 4-bit asynchronous and Synchronous counter (3 sessions)

****

Course Title: Mathematical Foundation of Computer Science - II
Course Code: COM - E3
Credits: 3
Marks: 75
Duration:45 Hrs

Prerequisite Courses: Nil

Course Objectives:

- To build mathematical foundations in the areas namely graph theory.
- Application of numerical analysis related to topics of computer science.

Course Outcomes:

On completion of the course students should be able to
CO1: System of linear equations in solving the problems
CO2: Apply the Interpolation methods for solving the problems numerically.
CO3: Demonstrate the process of curve fitting of data.
CO4: Determine the roots of polynomial equations.

SYLLABUS:

UNIT I: Systems of Linear Equations and Matrices, Linear Combinations and Linear Independence [10 HRS]

Systems of Linear Equations, Matrices and Elementary Row Operations, The Inverse of a Square Matrix, Matrix Equations, Applications of Systems of Linear Equations.
Linear Combinations and Linear Independence,

UNIT II: Vector Spaces, Linear Transformations, Eigenvalues and Eigenvectors [15 HRS]

Definition of a Vector Space, Subspaces, Basis and Dimension, Coordinates and Change of Basis
Linear Transformations, The Null Space and Range, Isomorphisms, Matrix Representation of Linear Transformations, Similarity.
Eigen values and Eigen vectors, Diagonalization

UNIT III: Interpolation, Numerical Integration, algebraic and transcendental equations [20 HRS]
Introduction; Various methods of interpolation; Various methods of curve fitting; Newton's method of forward interpolation formula; Newton's method of backward interpolation formula. Lagrange's formula. General quadrature formula; Trapezoidal rule; Simpson's one-third rule; Simpson's three-eighth rule; Weddle's rule. Graphical method; Bisection method; Method of false position; Secant method; Newton-Raphson method. Adjoint, inverse of a matrix; Rank; Linear equations; Characteristics roots and vectors.

REFERENCES:

MANDATORY:
1. Defranza, J., & Gagliardi, D. (2015). *Introduction to Linear Algebra with applications*. Waveland Press.
2. Parthasarathy.K,(1994) *Basic Graph Theory*, Tata McGraw-Hill Publishing
3. Goel,B., & Mittal,S. (1998) *Numerical Analysis*, PragatiPrakashan,
4. Iyengar,S.N.,(2010) *Matrices*, Anmol Publications.

SUPPLEMENTARY:

1. Clark, J., & Holton, D. A. (1991). *A first look at graph theory*. World Scientific.
2. Chatterjee,P. (1996) *Numerical Analysis*, RajhansPrakashanMandir.
3. Krishnamurthy,V. (1976) *Introduction to Linear Algebra*, Affiliated East-West Press.

WEB BASED:

1.https://www.cse.iitb.ac.in

Practicals : Mathematical Foundation of Computer Science - II
Credit: 1
Marks: 25
Duration: 30 Hrs

Linear equations
1. Systems of Linear Equations
2. Linear Transformations
3. Matrix Representation of Linear Transformations

Numerical Analysis
4) Find the value of dependent variable using Newton's forward formula for a given value of independent variable.
5) Use Newton's backward formula to estimate a value
6) Estimate a value using Lagrange's formula
7) Apply Simpson's three-eighth rule to find the value of integration
8)Apply Newton-Raphson method OR secant method to estimate the root of a equation

Linear Algebra
9) Find the rank of a matrix.
10) Find solutions of a system of equations
11) Find the Eigen values and Eigen vectors

****

Course Title: Web Designing
Course Code: COM-E4
Marks: 75
Credits: 3
Duration:45 Hrs

Prerequisite Courses: Nil

Course objectives:

- Design good user interfaces.
- Apply design principles such as learnability, visibility, error prevention, efficiency and graphic design.

Course Outcomes:

On completion of the course students will be able to:

CO1: Design Content for a web application.
CO2: Style content so as to provide an effective User Interface.
CO3: Provide for dynamism in the User Interface to enhance usability.
CO4: Develop a static web application.

SYLLABUS:

UNIT I: Structuring the UI with HTML/HTML5 [15 HRS]
User Interface – Introduction, its importance, design principles – learnability, visibility, error prevention, efficiency, graphic design. Design Patterns for GUI – View tree, Listener, Widget, Model-ViewController.

HTML - Introduction. The development process, basic HTML, formatting and fonts, commenting code, color, hyperlink, lists, tables, images, simple HTML forms, web site structure, Meta tags, Character entities, frames and frame sets.

HTML5 - Introduction, New Elements, Canvas, SVG, Drag/Drop, Geolocation, Video, Audio, Input types, form elements, form attributes, semantic, web storage, app cache, web workers, SSE

UNIT II : Styling the UI with CSS/CSS3 [10 HRS]
CSS - Introduction – Syntax, Id & Class, Backgrounds, Text, Fonts, Links, Lists, Tables. CSS Box Model – Border, Outline, Margin, Padding. Advanced - Grouping/Nesting, Dimension, Display, Positioning, Floating, Align, Pseudo-class, Pseudo-element, Navigation Bar, Image Gallery, Image Opacity, Image Sprites, Media Types, Attribute Selectors.
CSS3 - Introduction, Borders, Backgrounds, Gradients, Text Effects, Fonts, 2D Transforms, 3D Transforms, Transitions, Animations, Multiple Columns.

UNIT III: Dynamism in UI [20 HRS]
JavaScript - Introduction - What is JavaScript, Understanding Events, JavaScript Example, External JavaScript. Basic Elements – Comment, Variable, Global Variable, Data Types, Operators, If Statement, Switch, Loop: for and while, Function. JavaScript Objects – objects, Array. Browser Object Model - Browser Objects, Window Object, Document Object – getElementById, getElementsByName, getElementsByTagName, innerHTML property, inner Text property. Validation- form validation, email

validation.
JQuery : Introduction - Syntax, Selectors, Events. Effects- Hide/Show, Fade, Slide, Animate, stop(), Callback, Chaining. HTML/ CSS- Add, Remove, CSS Classes, css(), Dimensions, slider. Traversing – ancestors, descendants, siblings, filtering.

REFERENCES:

MANDATORY:

1. D.T,(2018) *Web Technologies*, Black Book,DreamTech

WEB BASED:

1.https://www.w3schools.com
2.https://www.tutorialspoint.com/html/index.htm
3.https://www.tutorialspoint.com/css/index.htm
4.https://www.tutorialspoint.com/javascript/index.htm
5.https://www.tutorialspoint.com/jquery/index.htm
6.https://www.udemy.com/courses/development/web-development/

Practicals : Web Designing
Marks: 25
Credits: 1
Duration: 30 Hrs

List of Assignments: (the numbers in brackets indicate number of practicals) :
1) Case studies to review UI designs [2 P]
2) Create a HTML page with the following: [3 P]
a) title heading paragraph emphasis strong and image elements
b) complex HTML table
c) simple HTML Form covering major form elements
d) Embed Video in an HTML page

3) Using CSS do the following: [3 P]
a) Create a Navigation bar (with dropdown) with CSS
b) Create a CSS Grid
c) Create a CSS3 based button
d) Make an image rounded shape
e) Create a CSS based sticky footer
f) Create CSS3 Corner Ribbon
g) Create CSS3 blurry text effect
h) Create CSS3 speech bubble shape
i) Create image cross fade with CSS3 transition
j) Set style for link hover active and visited states of hyperlink

4) Write JavaScript functions to : [4 P]
a) accept a string as a parameter and converts the first letter of each word of the string in upper case.
b) check whether a given credit card number is valid or not.
c) check whether a given value is an valid url or not.
d) check whether a given email address is valid or not.

e) print an integer with commas as thousands separators
f) remove items from a dropdown list.

5) Use JQuery to : [3 P]
a) Disable buttons
b) Make textbox read only
c) Uncheck check boxes
d) Confirm again
e) Sort
f) Switch rows and columns

A mini project combining all the technologies learnt using a front-end development framework such as bootstrap is recommended.

****

SEMESTER IV

Course Title: Computer Architecture and Organization
Course Code: Com-IV. C-6
Marks: 75
Credits: 3
Duration: 45Hours

Prerequisite Courses: Nil

Course Objectives:

- Conceptualize the basics of organizational and architectural issues of a digital computer.
- Analyze processor performance improvement using instruction level parallelism.
- Explain different types of addressing modes and memory organization.
- Study various data transfer techniques in digital computer.
- Learn microprocessor architecture and study assembly language programming.
- Understand the taxonomy of Parallel Processor.
- To have a thorough understanding of the basic structure and operation of a digital computer.

Course Outcomes:

At the end of the course students will be able to:

CO1: Identify various components of the Computer System.
CO2: Explain the detailed function of a typical microprocessor and its control unit.
CO3: Implement Assembly Language Program for 8086 processor for a given task.
CO4: Differentiate the function and role of semiconductor memories and map the cache memory for the given scenario.
CO5: Appraise the importance of input/output modules and Interrupts and their functions.
CO6: Distinguish the characteristics and function of I/O interfaces to computer system.
CO7: Illustrate the function of pipelined architecture and classify the Multiprocessor systems.

SYLLABUS

UNIT I: Computer System and Processor Unit: [15 HRS]
Function and structure of a computer, Interconnection of components, Performance of a computer. Computer Architecture – Princeton (Von Neumann) and Harvard architecture.Architecture of 8086 processor - Registers, ALU and Control unit, Data path in a CPU.Instruction cycle, Organization of a control unit – Block Diagram of Hardwired andMicroprogrammed control unit.

UNIT II: Instruction Set and Memory Subsystem: [15 HRS]
Machine instructions characteristics, Types of operations-data transfer, arithmetic,logical, conversion, I/O, system control, transfer of control; 8086 Instruction Set and Assembly language: Addressing modes-immediate, direct, indirect, register, register indirect, displacement, stack. Instruction formats - instruction length, allocation of bits,variable length instructions, Instruction set architectures – CISC and RISC architectures.Characteristics of memory system, the memory hierarchy, Semiconductor memories,Types of ROM & RAM, Cache memory unit - Concept of cache memory,Organization of a cache memory unit, Mapping methods, replacement algorithms,write policy, block size.

UNIT III: Input/Output Subsystem and Parallel Processing: [15 HRS]

General block diagram of External device & I/O module, Programmed I/O, Interrupt driven I/O, DMA, I/O channels and I/O processors. I/O interfaces – Serial port, Parallel port, PCI bus, SCSI bus, USB bus, Firewire and Infiniband.Classifications, Introduction to pipeline processing: Instruction pipeline & Arithmetic pipeline, Introduction to Array & Vector processors, Introduction to Multiprocessors.

REFERENCES:

MANDATORY:

1. Stallings, W. (2003). *Computer organization and architecture: designing for performance*. Pearson Education India..

SUPPLEMENTARY:

1. Mano, M. M. (2005). *Computer system architecture*. Dorling Kindesley Pearson.
2. Patterson, D. A., & Hennessy, J. L. (2013). *Computer organization and design MIPS edition: the hardware/software interface*. Newnes.
3. Douglas V. Hall, (2012), *Microprocessors and its Interfacing*, McGraw Hill Education (India) Private Limited.

WEB BASED:

1. http://williamstallings.com/ComputerOrganization/
2. https://www.tutorialspoint.com/computer_fundamentals/index.htm
3. http://www.ecs.umass.edu/ece/koren/architecture/
4. http://www.cs.colby.edu/djskrien/CPUSim/
5. https://teachcomputerscience.com/little-man-computer/
6. https://vivaxsolutions.com/web/lmc.aspx

Practicals: Computer Architecture and Organization
Credits: 1
Marks: 25
Duration: 30 Hrs

1. Study of Motherboard, Peripherals and the Computer System:O.S. Installation (Dual Boot): BIOS; Manage disk partitions: understand MBR-style partitions, (primary, extended, logical); list/create/delete partitions; Manage logical volumes: create/remove physical volumes, create/delete logical volumes, Boot loader.Installation of drivers; updating software packages
2. DOS Commands, Tools for Computer Management (Disk Management, Disk Cleanup, Defragmentation, Performance Monitor, System Restore etc).
Assembly language programs for 8086 using MASM / compatible assembler or Simulator, either in Windows or Linux.
3. Study of addressing modes.
4. Programs for arithmetic operations1
5. Programs for arithmetic operations2
6. Programs for arithmetic operations3
7. Programs for data transfer operations
8. Programs for logical operations1
9. Programs for logical operations2
10. Programs code conversion1
11. Programs code conversion2
12. Programs on sorting
13. Programs on searching
14. DOS/BIOS – Programming1

15. DOS/BIOS – Programming2
****

Course Title: Design and Analysis of Algorithms
Course Code : COM-E5
Marks :75
Credits :3
Duration :45 Hrs

Prerequisite Courses: Nil

Course Objectives:

- To study paradigms and approaches used to analyze and design algorithms and to appreciate the impact of algorithm design in practice.
- To ensure that students understand how the worst-case time complexity of an algorithm is defined, how asymptotic notation is used to provide a rough classification of algorithms and compare with one another, and how there are still some problems for which it is unknown whether there exist an efficient algorithm, and how to design efficient algorithms.

Course Outcomes:

At the end of the course students will be able to :

CO1: To analyze the performance of algorithms.
CO2: Choose appropriate algorithm design techniques for solving problems.
CO3: Analyze empirical results to get a deeper understanding of the algorithmic solutions.
CO4: Apply important algorithmic design paradigms and methods of analysis.

SYLLABUS:

UNIT I: Algorithm Analysis and Divide and Conquer Strategy [15 HRS]

What is an Algorithm?, Rules for writing Algorithms, Properties of Algorithms, Framework for design and analysis of algorithms(RAM model of computation),Recursive Algorithms, Space and Time Complexity by Tabular method(Performance Analysis) Elements of Divide and Conquer Algorithms, QuickSort algorithm, Merge sort analysis, Strassen's algorithm for matrix multiplication, Analysis of Binary Search,The Maximum subarray Problem.

UNIT II: Dynamic programming and Greedy Techniques [15 HRS]
General Method, caching v/s computation, Fibonacci numbers by recursion, Fibonacci numbers by cashing, Fibonacci numbers by dynamic programming,Optimal Binary Search Tree,Rod Cutting Problem. Elements of greedy strategy, Activity-selection problem, Job sequencing with deadlines.
Knapsack problem.

UNIT III: Graphs and Complexity Classes [15 HRS]
Elementary graph algorithms- Minimum spanning tree, growing a spanning tree, Kruskal and Prim algorithms, Breadth First search and Depth First Search, Travelling salesman problem. Introduction to polynomial time algorithms, NP, NP Complete, NP Hard, Approximation and Randomisation algorithms.

REFERENCES:

MANDATORY:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. MIT press.

SUPPLEMENTARY:

1.Horowitz.T, Sahni.S, Rajasekaran.S,(2010) "*Fundamentals of Computer Algorithms"*, Universities Press

Maryland,
2. A. Aho, J. Hopcroft, J. Ullman (2016).“*The Design and Analysis of Computer Algorithms*” Pearson Education India New Delhi,

WEB BASED:

1.http://www.iitk.ac.in/esc101/2009Jan/lecturenotes/timecomplexity
2. https://home.cse.ust.hk/~dekai/271/notes/L12/L12.pdf
3. https://nptel.ac.in/courses/106106131/
4. https://www.cs.cornell.edu/courses/cs3110/2012sp/lectures/lec20-master/lec20.html
5. https://www.cs.indiana.edu/~achauhan/Teaching/B401/LectureNotes/complexityClasses.html

Practicals: Design and Analysis of Algorithms
Credit : 1
Marks : 25
Duration: 30 Hrs

1 Program to find GCD of 2 numbers using Iterative approach and Recursive approach
2 Program for quickSort
3 Program for Mergesort
4 Program to perform Binary Search using Recursive approach
5 Program to implement maximum subarray problem.
6 Program to generate Fibonacci numbers using Dynamic Programming approach.
7 Program to implement Activity Selection Problem.
8 Program to implement job sequencing with Deadlines.
9 Program to implement Knapsack Problem
10 Program to implement Rod Cutting Problem.
11 Program to implement Binary Tree.
12 Program to represent graph using matrix/ Linked List.
13 Program to implement BFS/DFS Traversal on graph.
14 Program to implement Kruskal’s Algorithm
15 Program to implement Prim’s Algorithm

****

Course Title: Mobile Application Development
Course Code: COM-E10
Marks: 75
Credits: 3
Duration:45 Hrs

Prerequisite Courses: Nil

Course Objective:

- Students learn how to develop applications for mobile devices, including smart phones and tablets. Students are also introduced to the current mobile platforms, mobile application development environments and mobile device input methods. Students will design and build a variety of apps on a popular platform throughout the course to reinforce learning and to develop real competency.

Course Outcome :

At the end of the course students will be able to :

CO1: Explain mobile devices, including their capabilities and limitations.
CO2: Review current mobile platforms and their architectures.
CO3: Develop mobile applications on a popular mobile platform.
CO4: Evaluate development with another mobile platform.

SYLLABUS:

UNIT I:Introduction to mobile devices: [15 HRS]
Mobile devices vs. desktop devices, Why we Need Mobile App, Different Kinds of Mobile Apps, ARM and intel architectures, Power Management, Screen resolution, Touch interfaces,Application deployment - App Store, Google Play, Windows Store,Development environments – Android Studio, PhoneGAP,Native vs. web app, Browser-detection, Touch interfaces, Geolocation, Screen orientation,
Mobile browser ?interpretations? (Chrome/IE).Comparing and Contrasting architectures of Android, iOS and Windows, Underlying OS(Darwin vs. Linux vs. Windows), Kernel structure and native level programming,Runtime (Objective-C vs. Dalvik vs. WinRT), Security.Introduction to Android Operating System, Overview of android stack, Android features.Linux Kernel, Libraries, Android Runtime, Application Framework, Dalvik VM,Multithreading in Android-Async Task,Handler Post

UNIT II:Android Components: [15 HRS]
Activities, Services, Broadcast Receivers, Creating Broadcast receiver, Receiving System Broadcast, Understanding Broadcast action, category and data, Sending Broadcast. Content Providers, Views, layouts and Common UI components, Creating UI through code and XML,Activity life cycle, Intents-,Intent Filters, Intent-matching rules, Filters in your manifest.Communicating data among Activities. Selection components (Grid View, List View,Spinner), Adapters, Custom Adapters, Menus, Toast, Custom Toast, Dialogs, Status bar Notifications.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.

UNIT II:Data Storage: [15 HRS]
Shared Preferences, Android File System, Internal storage, External storage.
SQLite- Introducing SQLite, SQLiteOpenHelper and creating a database, Opening and closing a database, working with cursors, inserts, updates, and deletes.

Content Provider-Accessing built in content providers, Content provider MIME types, searching for content, Adding, changing, and removing content, creating content provider,Working with content files.

REFERENCES:

MANDATORY:

1. Lee, W. M. (2010). *Beginning iPad application development*. John Wiley & Sons.
2. Satya.K, Pro *Android 4* ; Dave MacLean (Apress)
3. Burnette, E. (2009). *Hello, Android introducing Google's mobile development platform 2nd*.
4. Cinar, O. (2012). *Android apps with Eclipse*. Apress.
5., Dimarzio, J.F. *Android- A Programmer'S Guide* (Tata McGraw Hill)

WEB BASED:

1. http://developer.android.com/index.html
2. http://www.appinventor.org/

Practicals : Mobile Application Development
Credit: 1
Marks: 25
Duration: 30 Hrs

1. Getting Started with Android – Installing the Development Environment, Configuring Android Stack
2. Creating the First Android Application - Creating a Simple Android Project, Debugging Application through DDMS. setting up environment. AVD Creation, Executing Project on Android Screen.
3. Android application development - Use of GUI components to implement a simple application such as a Calculator.
4. Review the earlier application making use of the advanced UI components.
5. Implementing Data storage application - an application to make Insert , update , Delete and retrieve operation on the database.

****

Paper Title: Server Side Programming
Paper Code: COM-E7
Marks: 75
Credits : 3
Duration:45 Hrs

Course Prerequisite: Nil

Course Objectives:

- Provide an in depth understanding of :
- Object oriented approaches to software development, in particular to the development of web applications.
- Server side scripts and their purpose

Course Outcomes:

On completion of the course students will be able to:

CO1:Design and implement server-side scripts.
CO2: Create and manipulate databases using SQL and server side technologies.
CO3: Create data documents using XML/JSON.
CO4: Enhancing web applications using AJAX and XML/JSON.
CO5: Develop dynamic web applications using the object oriented paradigm.

SYLLABUS:

UNIT I: Application of Object Oriented Approach to Software Development [10 HRS]
Review of Object Oriented Concepts - Class, Object, member variable, member function, Inheritance, Polymorphism, overloading, Data Abstraction, Encapsulation.Object oriented design and modeling.

UNIT II: Developing dynamic web applications with Server-side technologies [20HRS]
Static vs. Dynamic web pages, Need for Server Side technologies, Multitier Web Architecture. Common Gateway Interface standard, server-side includes, server APIs,server-side scripting – working principles, and implicit objects. Database and fileaccess. Comparison of Web servers.

UNIT III:Enhancing Web Applications with Ajax and XML/JSON [15HRS]
AJAX – introduction, purpose, advantages and disadvantages. Key elements of AJAX – introduction to XML. XML processing with server sidescript. XSL, transforms and templates. The XML Http Request object–methods and properties. Creating and using XML Http Request objects. Using XSLT with AJAX. JSON – Syntax, mixing literals, Array, object, encoding/decoding, JSON versus XML, server-side JSON tools.

Web Services - Introduction, its role. Basics – Creating, Publishing, Testing and Describing a Web services (WSDL)-Consuming a web service. SOAP – introduction, requests and responses. Role of UDDI – accessing registries. REST based webservices – building, deploying and consuming

REFERENCES:

MANDATORY:
1.Steelman, Murach'S *Java Servlets &amp*; JSP, 2/E, PHI

SUPPLEMENTARY:

1. Chappell.D, Jewell.T, "*Java Web Services : Using Java in Service Oriented Architectures*", O'Reilly
2. DT Editorial Services, (2018). *Web Technologies*, Black Book, DreamTech

WEB BASED:

1. https://docs.oracle.com/cd/E14571_01/web.1111/e13712/basics.htm#WBAPP117

Practicals: Server Side programming
Credit : 1
Marks : 25
Duration: 30 Hrs

List of suggested PRACTICALS (the numbers in brackets indicate number of practicals)
1) Perform OOAD of a given system using the following diagrams: [3 P]
a) use case diagram b) class diagram
2) Using server side programming and following OOAD principles develop a dynamic web application. [6 P]
3) Add AJAX and Web service(s) to the application. [3 P]

****

Course Title: Human Computer Interface
Course Code: COM-E8
Marks: 75
Credits: 03

Duration: 45 Hours

Pre-requisite course: Nil

Course Objectives:

- To study the different aspects of human computer interaction.
- To study computer interface design concepts.

Course Outcomes:

Upon completion of the course student will be able to:

CO 1: To understand the intricacies of human interaction with a computer System.
CO 2: To understand the concept of a graphical user interface, and its design characteristics.
CO 3: To recognize the human element its strengths and weakness for computer interaction.
CO 4: To know the principles of good screen design and layouts.
CO 5: To know the different navigation schemes on windows-based interface; learn the different types of selection devices and components of a window-based interface.
CO 6: To know the different types of interaction devices and media.

SYLLABUS:

UNIT I: Introduction to Human Computer Interaction: [10 HRS]
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.
Importance of: Human characteristics, Human consideration, Human interaction speeds,Understanding business functions. User centered design- Need-finding: Participant Observation,Interviewing, Additional Need finding, contextual inquiry & persona.

UNIT II: Rapid Prototyping and Graphical Interface Design: [20 HRS]
Rapid Prototyping: Story boarding. Paper Prototyping and Mockup, Video Prototyping, Creating and Comparing Alternatives.

Direct Manipulation. Mental Models. Heuristics (guidelines) for design.

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.

UNIT III: Heuristic Evaluation and Visualization [15 HRS]
Web user interface design – Jessy James Garette five layers of user experience.
Heuristic Evaluation: Heuristic Evaluation — Why and How?
Visualization, Amount of information, Focus and emphasis, Presentation information simply and meaningfully, Information retrieval on web, Statistical graphics.

REFERENCES:

Mandatory:
1. Cooper, A., Reimann, R., & Dubberly, H. (2003). *About face 2.0: The essentials of interaction design*. John

Wiley & Sons, Inc..
2. Alan.D, Janet.F, Gregory D. and Russell,B. (2012) *Human-Computer Interaction,* Prentice Hall.

Supplementary:

1. Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India.
2. Donald.A.N. (2010) *The Design of Everyday Things* Basic Books.

WEB BASED:

1. http://hcibib.org/
2. https://www.tutorialspoint.com/human_computer_interface/index.htm.
3.https://www.academia.edu/4955516/Wiley_The_Essential_Guide_to_User_Interface_Design_3rd_Edition_Apr_2007?auto=download.
4. https://www.slideshare.net/busaco/hci-2015-110-humancomputer-interaction-overview?qid=1c116f30-ec87-4eb4-a375-49b2bbe65d75&v=&b=&from_search=2

Practicals: Human Computer Interface
Credit : 1
Marks : 25
Duration: 30 Hrs

Suggested list of practical (Numbers in brackets indicate number of practicals)
1. Paper Prototyping using templates (1)
2. Conducting survey interview and summarizing the result(1)
3. Persona- conducting contextual interview and developing persona(1)
4. GUI design- form design, menu design, help, error messages(2)
5. Web UI design- pages, navigation, controls, Page submission – Asynchronous (2)
6. Report designs (2)
7. Visualization and info graphics (1)
8. Heuristic evaluation(2)
9. Story boarding (1)

****