Dr. B. R. AMBEDKAR UNIVERSITY-SRIKAKULAM

B.Sc. COMPUITER SCIENCE & IT SYLLABUS
STRUCTURE UNDER CHOICE BASED CREDITS SYSTEM
REVIEWED SYLLUBUS w.e.f. 2016-17

Structure of Syllabus

Semester	Part	Subject	Hrs.	Credits	IA	ES	Total
FIRST YEAR							
SEMESTER I	PART II	Computer Fundamentals and Photoshop	4	3	25	75	100
		Photo Shop Lab	2	2	0	50	50
SEMESTER II	PART II	Programming in C	4	3	25	75	100
		Programming in C Lab	2	2	0	50	50

Computer Fundamentals & Photoshop

Course Outcome
To explore basic knowledge on computers and Photoshop's beauty from the practical to the painterly artistic and to understand how Photoshop will help you create your own successful images

UNIT-I:
Introduction to computers, characteristics and limitations of computer, Block diagram of computer, types of computers, uses of computers, computer generations. Number systems :binary, hexa and octal numbering system

UNIT-II:
 Input and output devices: Keyboard and mouse, inputting data in other ways, Types of Software: system software, Application software, commercial, open source, domain and free ware software, Memories: primary, secondary and cache memory. Windows basics: desktop, start menu, icons.

Unit –III
Introduction to Adobe photoshop, Getting started with photoshop, creating and saving a document in photoshop, page layout and back ground, photoshop program window-title bar,menu bar,option bar,image window,image title bar,status bar,ruler,paletts,tool box,screen modes,saving files,reverting files,closing files.

Unit –IV
Images: working with images, image size and resolution ,image editing,colour modes and adjustments , Zooming & Panning an Image,, , Rulers, Guides & Grids- Cropping & Straightening an Image,image backgrounds ,making selections.
Working with tool box: working with pen tool, save and load selection-working with erasers-working with text and brushes-Colour manipulations: colour modes- Levels – Curves - Seeing Colour accurately - Patch tool – Cropping-Reading your palettes - Dust and scratches- Advanced Retouching- smoothing skin

Unit-V
Layers: Working with layers- layer styles- opacity-adjustment layers
Filters: The filter menu, Working with filters- Editing your photo shoot, presentation –how to create adds ,artstic filter,blur filter,brush store filter,distort filters,noice filters,pixelate filters,light effects,difference clouds,sharpen filters,printing.

Reference Books:

1. Fundamentals of Computers by Reema Thareja from Oxford University Press
2. Adobe Photoshop Class Room in a Book by Adobe Creative Team.
3. Photoshop: Beginner's Guide for Photoshop - Digital Photography, Photo Editing, Color Grading & Graphic...19 February 2016 by David Maxwell

Student Activity:
1. **Design a poster for technical paper presentation.**
2. **Create a digital scrap book.**

Photo Shop Lab

1. Create your Visiting card
2. Create Cover page for any text book
3. Create a Paper add for advertising of any commercial agency
4. Design a Passport photo
5. Create a Pamphlet for any program to be conducted by an organixation
6. Create Broacher for you college
7. Create Titles for any forthcoming film
8. Custom shapes creation
9. Create a Web template for your college
10. Convert color photo to black and white photo
11. Enhance and reduce the given Image size
12. Background changes
13. Design Box package cover
14. Design Texture and patterns
15. Filter effects & Eraser

PROGRAMMING IN C

Course Objectives

1. Learn how to solve common types of computing problems.
2. Learn data types and control structures of C
3. Learn to map problems to programming features of C.
4. Learn to write good portable C programs.

Course Outcomes

Upon successful completion of the course, a student will be able to:

1. Appreciate and understand the working of a digital computer
2. Analyze a given problem and develop an algorithm to solve the problem
3. Improve upon a solution to a problem
4. Use the 'C' language constructs in the right way
5. Design, develop and test programs written in 'C'

UNIT I

Introduction to Algorithms and Programming Languages: Algorithm – Key features of Algorithms – Some more Algorithms – Flow Charts – Pseudo code – Programming Languages – Generation of Programming Languages – Structured Programming Language-Design and Implementation of Correct, Efficient and Maintainable Programs.

Introduction to C: Introduction – Structure of C Program – Writing the first C Program – File used in C Program – Compiling and Executing C Programs – Using Comments – Keywords – Identifiers – Basic Data Types in C – Variables – Constants – I/O Statements in C- Operators in C- Programming Examples – Type Conversion and Type Casting

UNIT II

Decision Control and Looping Statements: Introduction to Decision Control Statements – Conditional Branching Statements – Iterative Statements – Nested Loops – Break and Continue Statement – Goto Statement

Functions: Introduction – using functions – Function declaration/ prototype – Function definition – function call – return statement – Passing parameters – Scope of variables – Storage Classes – Recursive functions – Type of recursion – Towers of Hanoi – Recursion vs Iteration

UNIT III

Arrays: Introduction – Declaration of Arrays – Accessing elements of the Array – Storing Values in Array – Calculating the length of the Array – Operations on Array – one dimensional array for inter-function communication – Two dimensional Arrays –Operations on Two Dimensional Arrays - Two Dimensional Arrays for inter-function communication – Multidimensional Arrays – Sparse Matrices

Strings: Introduction –Suppressive Input – String Taxonomy – String Operations – Miscellaneous String and Character functions

UNIT IV

Pointers: Understanding Computer Memory – Introduction to Pointers – declaring Pointer Variables – Pointer Expressions and Pointer Arithmetic – Null Pointers – Generic Pointers - Passing Arguments to Functions using Pointer – Pointer and Arrays – Passing Array to Function – Difference between Array Name and Pointer – Pointers and Strings – Array of pointers – Pointer and 2D Arrays – Pointer and 3D Arrays – Function Pointers – Array 0f Function Pointer – Pointers to Pointers – Memory Allocation in C Programs – Memory Usage – Dynamic Memory Allocation – Drawbacks of Pointers
Structure, Union, and Enumerated Data Types: Introduction – Nested Structures – Arrays of Structures – Structures and Functions – Self referential Structures – Union – Arrays of Unions Variables – Unions inside Structures – Enumerated Data Types

UNIT V

Files: Introduction to Files – Using Files in C – Reading Data from Files – Writing Data from Files – Detecting the End-of-file – Error Handling during File Operations – Accepting Command Line Arguments – Functions for Selecting a Record Randomly - Remove() – Renaming a File – Creating a Temporary File

REFERENCE BOOKS

1. Introduction to C programming by REEMA THAREJA from OXFORD UNIVERSITY PRESS
2. E Balagurusamy: ─COMPUTING FUNDAMENTALS & C PROGRAMMING – Tata McGraw-Hill, Second Reprint 2008, ISBN 978-0-07-066909-3.
3. Ashok N Kamthane: Programming with ANSI and Turbo C, Pearson Edition Publ, 2002.
4. Henry Mullish & Huubert L.Cooper: The Spirit of C An Introduction to modern Programming, Jaico Pub. House,1996.

Student Activity:
1. **Write a program for preparing the attendance particulars of students of your college at the end of semester according to following guidelines**
    a. **Above 75 % promoted**
    b. **Above 65% condoned**
    c. **Below 65% detained**
2. **Write a program for creating timetable or your class taking work load of faculty into consideration.**

PROGRAMMING IN C LAB

1. Find out the given number is perfect number or not using c program.
2. Write a C program to check whether the given number is Armstrong or not.
3. Write a C program to find the sum of individual digits of a positive integer.
4. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to print the Fibonacci series
5. Write a C program to generate the first n terms of the Fibonacci sequence.
6. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
7. Write a C program to find both the largest and smallest number in a list of integers.
8. Write a C program that uses functions to perform the following:
   a. Addition of Two Matrices
   b. Multiplication of Two Matrices
9. Write a program to perform various string operations
10. Write C program that implements searching of given item in a given list
11. Write a C program to sort a given list of integers in ascending order

Paper-III: OBJECT ORIENTED PROGRAMMING USING JAVA

Course Objectives

As the business environment becomes more sophisticated, the software development (software engineering is about managing complexity) is becoming increasingly complex. As of the best programming paradigm which helps to eliminate complexity of large projects, Object Oriented Programming (OOP) has become the predominant technique for writing software in the past decade. Many other important software development techniques are based upon the fundamental ideas captured by object-oriented programming.

Course Outcomes

At the end of this course student will:
1. Understand the concept and underlying principles of Object-Oriented Programming
2. Understand how object-oriented concepts are incorporated into the Java programming language
3. Develop problem-solving and programming skills using OOP concept
4. Understand the benefits of a well-structured program
5. Develop the ability to solve real-world problems through software development in high-level programming language like Java
6. Develop efficient Java applets and applications using OOP concept
7. Become familiar with the fundamentals and acquire programming skills in the Java language.

UNIT-1

FUNDAMENTALS OF OBJECT – ORIENTED PROGRAMMING: Introduction, Object Oriented paradigm, Basic Concepts of OOP, Benefits of OOP, Applications of OOP, Java features: **OVERVIEW OF JAVA LANGUAGE**: Introduction, Simple Java program structure, Java tokens, Java Statements, Implementing a Java Program, Java Virtual Machine, Command line arguments. **CONSTANTS, VARIABLES & DATA TYPES:** Introduction, Constants, Variables, Data Types, Declaration of Variables, Giving Value to Variables, Scope of variables, Symbolic Constants, Type casting, Getting Value of Variables, Standard Default values; **OPERATORS & EXPRESSIONS**.

UNIT-II

DECISION MAKING & BRANCHING: Introduction, Decision making with if statement, Simple if statement, if. Else statement, Nesting of if. else statements, the else if ladder, the switch statement, the conditional operator. **LOOPING**: Introduction, The While statement, the do-while statement, the for statement, Jumps in loops.

CLASSES, OBJECTS & METHODS: Introduction, Defining a class, Adding variables, Adding methods, Creating objects, Accessing class members, Constructors, Method overloading, Static members, Nesting of methods;

UNIT-III

INHERITANCE: Extending a class, Overloading methods, Final variables and methods, Final classes, Abstract methods and classes;

ARRAYS, STRINGS AND VECTORS: Arrays, One-dimensional arrays, Creating an array, Two – dimensional arrays, Strings, Vectors, Wrapper classes;

INTERFACES: MULTIPLE INHERITANCE: Introduction, Defining interfaces, Extending interfaces, Implementing interfaces, Assessing interface variables;

UNIT-IV

MULTITHREADED PROGRAMMING: Introduction, Creating Threads, Extending the Threads, Stopping and Blocking a Thread, Lifecycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the 'Runnable' Interface.

MANAGING ERRORS AND EXCEPTIONS: Types of errors: Compile-time errors, Run-time errors, Exceptions, Exception handling, Multiple Catch Statements, Using finally statement,

UNIT-V

APPLET PROGRAMMING: local and remote applets, Applets and Applications, Building Applet code, Applet Life cycle: Initialization state, Running state, Idle or stopped state, Dead state, Display state.

PACKAGES: Introduction, Java API Packages, Using System Packages, Naming conventions, Creating Packages, Accessing a Package, using a Package.

MANAGING INPUT/OUTPUT FILES IN JAVA: Introduction, Concept of Streams, Stream classes, Byte Stream Classes, Input Stream Classes, Output Stream Classes, Character Stream classes: Reader stream classes, Writer Stream classes, Using Streams, Reading and writing files.

Reference Books:

1. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill

   Company.

2. John R. Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TATA McGraw-Hill Company.

3. Deitel &Deitel. Java TM: How to Program, PHI (2007)

4. Java Programming: From Problem Analysis to Program Design- D.S Mallik

5. Object Oriented Programming Through Java by P. Radha Krishna, Universities Press (2008)

Student Activity:
1. Create a front end using JAVA for the student database created
2. Learn the difference between ODBC and JDBC

OBJECT ORIENTED PROGRAMMING USING JAVA LAB

1. Write a program to perform various String Operations
2. Write a program on class and object in java
3. Write a program to illustrate Function Overloading & Function Overriding methods in Java
4. Write a program to illustrate the implementation of abstract class
5. Write a program to implement Exception handling
6. Write a program to create packages in Java
7. Write a program on interface in java
8. Write a program to Create Multiple Threads in Java
9. Write a program to Write Applets to draw the various polygons
10. Write a program which illustrates the implementation of multiple Inheritance using interfaces in Java
11. Write a program to assign priorities to threads in java

Paper-IV: DATA STRUCTURES

Course Objectives

To introduce the fundamental concept of data structures and to emphasize the importance of data structures in developing and implementing efficient algorithms..

Course Outcomes

After completing this course satisfactorily, a student will be able to:
1. Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
2. Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs.
3. Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
4. Demonstrate different methods for traversing trees
5. Compare alternative implementations of data structures with respect to performance
6. Compare and contrast the benefits of dynamic and static data structures implementations
7. Describe the concept of recursion, give examples of its use, describe how it can be implemented using a stack.
8. Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing.

UNIT I
Concept of Abstract Data Types (ADTs) - Data Types, Data Structures, Storage Structures, and File Structures, Primitive and Non-primitive Data Structures, Linear and Non-linear Data Structures.
Linear Lists – ADT, Array and Linked representations, Pointers.
Arrays – ADT, Mappings, Representations, Sparse Matrices,   Sets – ADT, Operations
Linked Lists: Single Linked List, Double Linked List, Circular Linked List, applications

UNIT II
Stacks: Definition, ADT, Array and Linked representations, Implementations and Applications
Queues: Definition, ADT, Array and Linked representations, Circular Queues, Dequeues, Priority Queues, Implementations and Applications.

UNIT III
Trees: Binary Tree, Definition, Properties, ADT, Array and Linked representations, Implementations and Applications. Binary Search Trees (BST) – Definition, ADT, Operations and Implementations, BST Applications. Threaded Binary Trees, Heap trees.

UNIT IV
Graphs – Graph and its Representation, Graph Traversals, Connected Components, Basic Searching Techniques, Minimal Spanning Trees

UNIT- V
Sorting and Searching: Selection, Insertion, Bubble, Merge, Quick, Heap sort, Sequential and Binary Searching.

REFERENCE BOOKS

1. D S Malik, Data Structures Using C++, Thomson, India Edition 2006.
2. Sahni S, Data Structures, Algorithms and Applications in C++, McGraw-Hill, 2002.
3. SamantaD, Classic Data Structures, Prentice-Hall of India, 2001.
4. Heilman G I,. Data Structures and Algorithms with Object-Oriented Programming, Tata McGraw-l lill. 2002. (Chapters I and 14).
5. Tremblay P, and Sorenson P G, Introduction to Data Structures with Applications, Tata McGraw-Hill,

Student activity:

1. **Create a visible stack using C-graphics**
2. **Create a visible Queue using C-graphics**

DATA STRUCTURES USING JAVA LAB

1. Write a Program to implement the Linked List operations

2. Write a Program to implement the Stack operations using an array.

3. Write Programs to implement the Queue operations using an array.

4. Write Programs to implement the Stack operations using a singly linked list.

5. Write Programs to implement the Queue operations using a singly linked list.

6. Write a program for arithmetic expression evaluation

7. Write a program to implement Double Ended Queue using a doubly linked list.

8. Write a program to search an item in a given list using Linear Search and Binary Search

9. Write a program for Quick Sort

10. Write a program for Merge Sort

11. Write a program on Binary Search Tree operations(insertion, deletion and traversals)

12. Write a program for Graph traversals

Dr. B. R. AMBEDKAR UNIVERSITY-SRIKAKULAM

B.Sc. COMPUITER SCIENCE & IT SYLLABUS
STRUCTURE UNDER CHOICE BASED CREDITS SYSTEM
REVIEWED SYLLUBUS w.e.f. 2016-17

Structure of Syllabus

Semester	Part	Subject	Hrs.	Credits	IA	ES	Total
SECOND YEAR							
SEMESTER III	PART II	Object Oriented Programming Using Java	4	3	25	75	100
		Object Oriented Programming Using Java Lab	2	2	0	50	50
SEMESTER IV	PART II	Data Structures	4	3	25	75	100
		Data Structures using Java Lab	2	2	0	50	50

II YEAR III SEMESTER

OBJECT ORIENTED PROGRAMMING USING JAVA

Course Objectives

As the business environment becomes more sophisticated, the software development (software engineering is about managing complexity) is becoming increasingly complex. As of the best programming paradigm which helps to eliminate complexity of large projects, Object Oriented Programming (OOP) has become the predominant technique for writing software in the past decade. Many other important software development techniques are based upon the fundamental ideas captured by object-oriented programming.

Course Outcomes

At the end of this course student will:
1. Understand the concept and underlying principles of Object-Oriented Programming
2. Understand how object-oriented concepts are incorporated into the Java programming language
3. Develop problem-solving and programming skills using OOP concept
4. Understand the benefits of a well structured program
5. Develop the ability to solve real-world problems through software development in high-level programming language like Java
6. Develop efficient Java applets and applications using OOP concept
7. Become familiar with the fundamentals and acquire programming skills in the Java language.

UNIT-1

FUNDAMENTALS OF OBJECT – ORIENTED PROGRAMMING: Introduction, Object Oriented paradigm, Basic Concepts of OOP, Benefits of OOP, Applications of OOP, Java features: **OVERVIEW OF JAVA LANGUAGE**: Introduction, Simple Java program structure, Java tokens, Java Statements, Implementing a Java Program, Java Virtual Machine, Command line arguments. **CONSTANTS, VARIABLES & DATA TYPES:** Introduction, Constants, Variables, Data Types, Declaration of Variables, Giving Value to Variables, Scope of variables, Symbolic Constants, Type casting, Getting Value of Variables, Standard Default values; **OPERATORS & EXPRESSIONS**.

UNIT-II

DECISION MAKING & BRANCHING: Introduction, Decision making with if statement, Simple if statement if. Else statement, Nesting of if. else statements, the else if ladder, the switch statement, the conditional operator. **LOOPING**: Introduction, The While statement, the do-while statement, the for statement, Jumps in loops.

CLASSES, OBJECTS & METHODS: Introduction, Defining a class, Adding variables, Adding methods, Creating objects, Accessing class members, Constructors, Method overloading, Static members, Nesting of methods;

UNIT-III

INHERITANCE: Extending a class, Overloading methods, Final variables and methods, Final classes, Abstract methods and classes;

ARRAYS, STRINGS AND VECTORS: Arrays, One-dimensional arrays, Creating an array, Two – dimensional arrays, Strings, Vectors, Wrapper classes;

INTERFACES: MULTIPLE INHERITANCE: Introduction, Defining interfaces, Extending interfaces, Implementing interfaces, Assessing interface variables;

UNIT-IV

MULTITHREADED PROGRAMMING: Introduction, Creating Threads, Extending the Threads, Stopping and Blocking a Thread, Lifecycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the 'Runnable' Interface.

MANAGING ERRORS AND EXCEPTIONS: Types of errors: Compile-time errors, Run-time errors, Exceptions, Exception handling, Multiple Catch Statements, Using finally statement,

UNIT-V

APPLET PROGRAMMING: local and remote applets, Applets and Applications, Building Applet code, Applet Life cycle: Initialization state, Running state, Idle or stopped state, Dead state, Display state.

PACKAGES: Introduction, Java API Packages, Using System Packages, Naming conventions, Creating Packages, Accessing a Package, using a Package.

MANAGING INPUT/OUTPUT FILES IN JAVA: Introduction, Concept of Streams, Stream classes, Byte Stream Classes, Input Stream Classes, Output Stream Classes, Character Stream classes: Reader stream classes, Writer Stream classes, Using Streams, Reading and writing files.

Reference Books:

1. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company.
2. John R. Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TATA McGraw-Hill Company.
3. Deitel &Deitel. Java TM: How to Program, PHI (2007)
4. Java Programming: From Problem Analysis to Program Design- D.S Mallik
5. Object Oriented Programming Through Java by P. Radha Krishna, Universities Press (2008)

Student Activity:
1. Create a front end using JAVA for the student database created
2. Learn the difference between ODBC and JDBC

OBJECT ORIENTED PROGRAMMING USING JAVA LAB

1. Write a program to perform various String Operations
2. Write a program on class and object in java
3. Write a program to illustrate Function Overloading & Function Overriding methods in Java
4. Write a program to illustrate the implementation of abstract class
5. Write a program to implement Exception handling
6. Write a program to create packages in Java
7. Write a program on interface in java
8. Write a program to Create Multiple Threads in Java
9. Write a program to Write Applets to draw the various polygons
10. Write a program which illustrates the implementation of multiple Inheritance using interfaces in Java
11. Write a program to assign priorities to threads in java

II YEAR IV SEMESTER

DATA STRUCTURES

Course Objectives

To introduce the fundamental concept of data structures and to emphasize the importance of data structures in developing and implementing efficient algorithms..

Course Outcomes

After completing this course satisfactorily, a student will be able to:
1. Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
2. Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs.
3. Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
4. Demonstrate different methods for traversing trees
5. Compare alternative implementations of data structures with respect to performance
6. Compare and contrast the benefits of dynamic and static data structures implementations
7. Describe the concept of recursion, give examples of its use, describe how it can be implemented using a stack .
8. Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing.

UNIT I
Concept of Abstract Data Types (ADTs)- Data Types, Data Structures, Storage Structures, and File Structures, Primitive and Non-primitive Data Structures, Linear and Non-linear Data Structures.
Linear Lists – ADT, Array and Linked representations, Pointers.
Arrays – ADT, Mappings, Representations, Sparse Matrices, Sets – ADT, Operations
Linked Lists: Single Linked List, Double Linked List, Circular Linked List , applications

UNIT II
Stacks: Definition, ADT, Array and Linked representations, Implementations and Applications
Queues: Definition, ADT, Array and Linked representations, Circular Queues, Dequeues, Priority Queues, Implementations and Applications.

UNIT III
Trees: Binary Tree, Definition, Properties, ADT, Array and Linked representations, Implementations and Applications. Binary Search Trees (BST) – Definition, ADT, Operations and Implementations, BST Applications. Threaded Binary Trees, Heap trees.

UNIT IV
Graphs – Graph and its Representation, Graph Traversals, Connected Components, Basic Searching Techniques, Minimal Spanning Trees
UNIT- V

Sorting and Searching: Selection, Insertion, Bubble, Merge, Quick, Heap sort, Sequential and Binary Searching.

REFERENCE BOOKS

1. D S Malik, Data Structures Using C++, Thomson, India Edition 2006.
2. Sahni S, Data Structures, Algorithms and Applications in C++, McGraw-Hill, 2002.
3. SamantaD, Classic Data Structures, Prentice-Hall of India, 2001.
4. Heilman G I,. Data Structures and Algorithms with Object-Oriented Programming, Tata McGraw-l lill. 2002. (Chapters I and 14).
5. Tremblay P, and Sorenson P G, Introduction to Data Structures with Applications, Tata McGraw-Hill,

Student activity:

1. **Create a visible stack using C-graphics**
2. **Create a visible Queue using C-graphics**

DATA STRUCTURES USING JAVA LAB

1. Write a Program to implement the Linked List operations

2. Write a Program to implement the Stack operations using an array.

3. Write Programs to implement the Queue operations using an array.

4. Write Programs to implement the Stack operations using a singly linked list.

5. Write Programs to implement the Queue operations using a singly linked list.

6. Write a program for arithmetic expression evaluation

7. Write a program to implement Double Ended Queue using a doubly linked list.

8. Write a program to search an item in a given list using Linear Search and Binary Search

9. Write a program for Quick Sort

10. Write a program for Merge Sort

11. Write a program on Binary Search Tree operations(insertion, deletion and traversals)

12. Write a program for Graph traversals

Dr. B. R. AMBEDKAR UNIVERSITY-SRIKAKULAM

III B.Sc. (COMPUTER SCIENCE) SYLLABUS
STRUCTURE UNDER CHOICE BASED CREDITS SYSTEM
REVIEWED SYLLUBUS w.e.f. 2016-17

SEMESTER V

THIRD YEAR						
SEMESTER V	V	DBMS	3	3	25	75	100
		DBMS Lab	2	2	0	50	50
	VI	Software Engineering	3	3	25	75	100
		Software Engineering Lab	2	2	0	50	50

Paper-V: Data Base Management System

Course Objective:

Design & develop database for large volumes & varieties of data with optimized data processing techniques.

Course Outcomes

On completing the subject, students will be able to:
 1. Design and model of data in database.
 2. Store, Retrieve data in database.

UNIT I

Overview of Database Management System: Introduction, file-based system, Drawbacks of file-Based System ,Data and information, Database, Database management System, Objectives of DBMS, Evaluation of Database management System, Classification of Database Management System, DBMS Approach, advantages of DBMS, Anis/spark Data Model, data models, Components and Interfaces of Database Management System. Database Architecture, Situations where DBMS is not Necessary, DBMS Vendors and Their Products.

UNIT II

Entity-Relationship Model: Introduction, the building blocks of an entity relationship diagram, classification of entity sets, attribute classification, relationship degree, relationship classification, reducing ER diagram to tables, enhanced entity-relationship model (EER model), generalization and specialization, **IS A** relationship and attribute inheritance, multiple inheritance, constraints on specialization and generalization, aggregation and composition, entity clusters, connection types, advantages of ER modeling.

UNIT III
Relational Model: Introduction, CODD Rules, relational data model, concept of key, relational integrity, relational algebra, relational algebra operations, advantages of relational algebra, limitations of relational algebra, relational calculus, tuple relational calculus,domain relational Calculus (DRC). QBE

UNIT IV

Structured Query Language: Introduction, History of SQL Standard, Commands in SQL, Data Types in SQL, Data Definition Language, Selection Operation, Projection Operation, Aggregate functions, Data Manipulation Language, Table Modification Commands, Table Truncation, Imposition of Constraints, Join Operation, Set Operation,
View, Sub Query, Embedded SQL,

UNIT V

PL/SQL: Introduction, Shortcoming in SQL, Structure of PL/SQL, PL/SQL Language Elements, Data Types, Operators Precedence, Control Structure, Steps to Create a PL/SQL, Program, Iterative Control, Cursors, Steps to create a Cursors, Procedure, Function, Packages, Exceptions Handling, Database Triggers, Types of Triggers.

Reference Books

1. "Database System Concepts" by Abraham Silberschatz, Henry Korth, and S. Sudarshan, McGrawhill, 2010, 9780073523323
2. "Database Management Systems" by Raghu Ramakrishnan, McGrawhill, 2002,
3. Fundamentals of Relational Database Management Systems by S. Sumathi, S. Esakkirajan, Springer Publications
4. "An Introduction to Database Systems" by Bipin C Desai
5. "Principles of Database Systems" by J. D. Ullman
6. "Fundamentals of Database Systems" by R. Elmasri and S. Navathe

Student Activity:
1. Create your college database for placement purpose.
2. Create faculty database of your college with their academic performance scores

III YEAR V SEMESTER

DATABASE MANAGEMENT SYSTEMS LAB

1. Draw ER diagrams for train services in a railway station
2. Draw ER diagram for hospital administration
3. Creation of college database and establish relationships between tables
4. Write a view to extract details from two or more tables
5. Write a stored procedure to process students results
6. Write a program to demonstrate a function
7. Write a program to demonstrate blocks, cursors & database triggers.
8. Write a program to demonstrate Joins
9. Write a program d
10. Write a program to demonstrate of   Aggregate functions
11. Creation of Reports based on different queries
12. Usage of file locking table locking, facilities in applications.

Paper VI: **Software Engineering**

Course Objectives

The Objective of the course is to assist the student in understanding the basic theory of software engineering, and to apply these basic theoretical principles to a group software development project.

Course outcomes
1. Ability to gather and specify requirements of the software projects.
2. Ability to analyze software requirements with existing tools
3. Able to differentiate different testing methodologies
4. Able to understand and apply the basic project management practices in real life projects
5. Ability to work in a team as well as independently on software projects

UNIT I

INTRODUCTION: Software Engineering Process paradigms - Project management - Process and Project Metrics – software estimation - Empirical estimation models - Planning - Risk analysis - Software project scheduling.

UNIT II
REQUIREMENTS ANALYSIS: Requirement Engineering Processes – Feasibility Study – Problem of Requirements – Software Requirement Analysis – Analysis Concepts and Principles – Analysis Process – Analysis Model

UNIT III
SOFTWARE DESIGN: Software design - Abstraction - Modularity - Software Architecture - Effective modular design - Cohesion and Coupling - Architectural design and Procedural design - Data flow oriented design.

UNIT IV
USER INTERFACE DESIGN AND REAL TIME SYSTEMS: User interface design - Human factors - Human computer interaction - Human - Computer Interface design - Interface design - Interface standards.

UNIT V
SOFTWARE QUALITY AND TESTING: Software Quality Assurance - Quality metrics - Software Reliability - Software testing - Path testing – Control Structures testing - Black Box testing - Integration, Validation and system testing - Reverse Engineering and Re-engineering. CASE tools –projects management, tools - analysis and design tools – programming tools - integration and testing tool - Case studies.

REFERENCE BOOKS:
1. Roger Pressman S., "Software Engineering: A Practitioner's Approach", 7th
Edition, McGraw Hill, 2010.
2. Software Engineering Principles and Practice by Deepak Jain Oxford University Press
2. Sommerville, "Software Engineering", Eighth Edition, Pearson Education, 2007
3. Pfleeger, "Software Engineering: Theory & Practice", 3rd Edition, Pearson
Education, 2009
4. Carlo Ghazi, Mehdi Jazayari, Dino Mandrioli, "Fundamentals of Software
Engineering", Pearson Education, 2003

Student Activity:

1. **Visit any financial organization nearby and prepare requirement analysis report**
2. **Visit any industrial organization and prepare risk chart.**

Software Engineering Lab

1. Studying various phases of Water-Fall Model.

2. Prepare SRS for Banking or On line book store domain problem

3. Using COCOMO model estimate effort for Banking or on line book store domain

problem.

4. Calculate effort using FP oriented estimation model

5. Analyze the Risk related to the project and prepare RMMM plan.

6. Develop Time-line chart and project table using PERT or CPM project scheduling methods.

7. Draw E-R diagram, DFD, CFD and STD for the project.

8. Design of the test cases.

9. Prepare FTR. Version control and change control for software configuration item

Dr. B. R. AMBEDKAR UNIVERSITY-SRIKAKULAM
III B.Sc. (Computer Science/Information Technology) SYLLABUS
STRUCTURE UNDER CHOICE BASED CREDITS SYSTEM
REVIEWED SYLLUBUS w.e.f. 2016-17

Structure of Computer Science/Information Technology (IT) Syllabus

Semester	Paper	Subject	Hrs.	Credits	IA	ES	Total
		FIRST YEAR					
SEMESTER I	I	Computer Fundamentals and Photoshop	4	3	25	75	100
		Photo Shop Lab	2	2	0	50	50
SEMESTER II	II	Programming in C	4	3	25	75	100
		Programming in C Lab	2	2	0	50	50
		SECOND YEAR					
SEMESTER III	III	Object Oriented Programming Using Java	4	3	25	75	100
		Object Oriented Programming Using Java Lab	2	2	0	50	50
SEMESTER IV	IV	Data Structures	4	3	25	75	100
		Data Structures using Java Lab	2	2	0	50	50
		THIRD YEAR					
SEMESTER V	V	DBMS	3	3	25	75	100
		DBMS Lab	2	2	0	50	50
	VI	Software Engineering	3	3	25	75	100
		Project- 1	2	2	0	50	50
SEMESTER VI	VII (A/B/C)	**Elective-I**					
		A. Operating Systems	3	3	25	75	100
		Operating Systems Lab	3	2	0	50	50
		B. Computer Networks	3	3	25	75	100
		Computer Networks Lab	3	2	0	50	50
		C. Web Technologies	3	3	25	75	100
		Web Technologies Lab	3	2	0	50	50
	VIII Cluster – A-A1,A2 or Cluster-B-B1,B2 Or Cluster – C – C1,C2	**Elective-II(Cluster A)**					
		A1.Foundations of Data Science	3	3	25	75	100
		Foundations of Data Science Lab (through R)	3	2	0	50	50
		A2.Big Data Technology	3	3	25	75	100
		Big Data Technology Lab (Hadoop)	3	2	0	50	50
		Elective-II(Cluster B)					
		B1. Distributed Systems	3	3	25	75	100
		Distributed Systems Lab	3	2	0	50	50
		B2. Cloud Computing	3	3	25	75	100
		Cloud Computing Lab	3	2	0	50	50
		Elective-II(Cluster C)					
		C1. PHP – MySQL & WordPress	3	3	25	75	100
		PHP-MySQL & Wordpress Lab	3	2	0	50	50
		C2. Advanced JavaScript : JQuery, Ajax, Angular JS & JSON	3	3	25	75	100
		Advanced JavaScript Lab	3	2	0	50	50
		Project – 2	5	5	25	75	100

III YEAR VI SEMESTER

Paper-VII: Elective-A

Operating Systems

Course Objectives

1. To understand the services provided by and the design of an operating system.
2. To understand the structure and organization of the file system.
3. To understand what a process is and how processes are synchronized and scheduled.
4. To understand different approaches to memory management.
5. Students should be able to use system calls for managing processes, memory and the file system.

Course Outcomes

1. Analyze the concepts of processes in operating system and illustration of the scheduling of processor for a given problem instance.
2. Identify the dead lock situation and provide appropriate solution so that protection and security of the operating system is also maintained.
3. Analyze memory management techniques, concepts of virtual memory and disk scheduling.
4. Understand the implementation of file systems and directories along with the interfacing of IO devices with the operating system.

UNIT - I

Operating System Introduction: Operating Systems Objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time shared, Parallel, Distributed Systems, Real-Time Systems, Operating System services.

UNIT - II

Process and CPU Scheduling - Process concepts - The Process, Process State, Process Control Block, Threads, Process Scheduling - Scheduling Queues, Schedulers, Context Switch, Preemptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms, Case studies: Linux, Windows.

Process Coordination - Process Synchronization, The Critical section Problem, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT - III

Memory Management and Virtual Memory - Logical & physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table. Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Performance of Demanding Paging, Page Replacement Page Replacement Algorithms, Allocation of Frames.

UNIT - IV

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Structure,

Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling.

UNIT - V

Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock.

.

REFERENCES BOOKS:

1. Operating System Principles, Abraham Silberchatz, Peter B. Galvin, Greg Gagne 8th Edition, Wiley Student Edition.
2. Principles of Operating Systems by Naresh Chauhan, OXFORD University Press
3. Operating systems - Internals and Design Principles, W. Stallings, 6th Edition, Pearson.
4. Modern Operating Systems, Andrew S Tanenbaum 3rd Edition PHI.
5. Operating Systems A concept - based Approach, 2nd Edition, D. M. Dhamdhere, TMH.
6. Principles of Operating Systems, B. L. Stuart, Cengage learning, India Edition.
7. Operating Systems, A. S. Godbole, 2nd Edition, TMH

Student Activity:
1. Load any new operating system into your computer.
2. Partition the memory in your system
3. Create a semaphore for process synchronization

Paper-VII: Elective-A

Operating Systems Lab

Objectives:

- To use linux operating system for study of operating system concepts.
- To write the code to implement and modify various concepts in operating systems

Outcomes:

- The course objectives ensure the development of students applied skills in operating systems related areas.
- Students will gain knowledge in writing software routines modules or implementing various concepts of operating system.

List of Experiments:

1. 1.Usage of following commands
   Ls,pwd,tty,cat,who,who am I,rm, mkdir,rmdir,touch,cd.
2. Usage of following commands
   Cal,cat(append),cat(concatenate),mv,cp,man,date.
3. Usage of following commands
   Chmod,grep,tput(clear,highlight),bc.
4. Write a shell script to check if the number entered at the command line is Prime or not.
5. Write a shell script to modify "cal" command to display calendars of the specified months.
6. Write a shell script to modify "cal" command to display calendars of the specified range of months.
7. Write a shell script to accept a login name. If not a valid login name display message "entered login name is invalid"
8. Write a shell script to display date in the mm/dd/yy format.
9. To implement the FCFS Algorithm.
10. To implement the shortest job First Algorithm.
11. To implement the priority algorithm.
12. To implement the round robin Algorithm.
13. To implement the FIFO page replacement algorithm

14. To implement the LRU page replacement Algorithm.
14. To implement the Resource request Algorithm.
15. To implement the First-Fit, Best-Fit, Worst-Fit Algorithm.
16. To implement the sequential file organization.
17. To implement the Random file organization

18. Simulate Page Replacement Algorithms FIFO
19. Simulate Page Replacement Algorithms LRU

21. Simulate Page Replacement Algorithms OPTIMAL
22. Simulate Algorithm For Deadlock Prevention

III YEAR VI SEMESTER

Paper-VII: Elective-B

COMPUTER NETWORKS

Course Objectives

1. To provide an introduction to the fundamental concepts on data communication and the design of computer networks.
2. To get familiarized with the basic protocols of computer networks.

Course Outcomes
After this course, the student will be able to
1. Identify the different components in a Communication System and their respective roles.
2. Describe the technical issues related to the local Area Networks
3. Identify the common technologies available in establishing LAN infrastructure.

UNIT – I

Introduction: Uses of Computer Networks, Network Hardware, Network Software, Reference Models, Example Networks.

The Physical Layer: The Theoretical Basis for Data Communication, Guided Transmission Media, Wireless transmission, the public switched telephone network

UNIT – II

The Data Link Layer: Data Link Layer Design Issues, Error Detection and Correction, Sliding Window Protocols.

The Medium Access Control Sub-layer: The channel allocation problem, **Multiple Access Protocols**, **Ethernet,** Data Link Layer Switching.

UNIT – III

The Network Layer: Network Layer Design Issues, Routing Algorithms, Congestion control algorithms, Quality of Service.

Internet Working, The Network Layer in the Internet

UNIT – IV:

The Transport Layer: The Transport Service, Elements of Transport Protocols, Congestion Control Algorithms, The Internet Transport Protocols, The Internet Transport Protocols: TCP, Delay Tolerant Networks.

UNIT – V:

The Application Layer: DNS – The Domain Name System, Electronic Mail, The World Wide Web, Real Time Audio & Video, Content Delivery & Peer-to-Peer.

Reference Books:

1. Andrew S. Tanenbaum, "Computer Networks", Fifth Edition, Pearson Education.

2. Bhushan Trivedi, Computer Networks , Oxford University Press

3. James F.Kurose, Keith W.Ross, "Computer Networking", Third Edition, Pearson Education

4. Behrouz A Forouzan, "Data Communications and Networking", Fourth Edition, TMH (2007).

5. Kurose & Ross, "*COMPUTER NETWORKS*" – A Top-down approach featuring the Internet", Pearson Education – Alberto Leon – Garciak.

Student Activity:

1. **Study the functioning of network devices available in your organization .**
2. **Prepare a pictorial chart of LAN connections in your organization**

Paper-VII: Elective-B

COMPUTER NETWORKS LAB

OBJECTIVES:
 1. Analyze the different layers in networks.
2. Define, use, and differentiate such concepts as OSI-ISO,TCP/IP.
 3. How to send bits from physical layer to data link layer
 4. Sending frames from data link layer to Network layer
5. They can understand how the data transferred from source to destination
 6. They can come to know that how the routing algorithms worked out in network layer

List of Experiments:
1. Analyze the different layers in networks.
2. Define, use, and differentiate such concepts as OSI-ISO,TCP/IP.

List of Experiments:

1. Write a program to implement data link layer framing method bit stuffing.

2. Write a program to implement data link layer framing method character stuffing.

3. Write a program to implement data link layer framing method character count.

4. Write a program to implement Cyclic Redundancy Check (CRC 12, CRC 16 and CRC

CCIR) on a data set of characters.

5. Write a program to implement Dijkstra's algorithm to compute the shortest path through

a graph.

6. Write a program to implement subnet graph with weights indicating delay between

7. Write a program to implement subnet

Paper-VII: Elective-C
Web Technologies

Course Objective

To provide knowledge on web architecture, web services, client side and server side scripting technologies to focus on the development of web-based information systems and web services.

To provide skills to design interactive and dynamic web sites.

Course Outcome

1. To understand the web architecture and web services.
2. To practice latest web technologies and tools by conducting experiments.
3. To design interactive web pages using HTML and Style sheets.
4. To study the framework and building blocks of .NET Integrated Development Environment.
5. To provide solutions by identifying and formulating IT related problems.

UNIT – I

HTML: Basic HTML, Document body, Text, Hyper links, adding more formatting, Lists, Tables using images. More HTML: Multimedia objects, Frames, Forms towards interactive, HTML document heading detail

UNIT – II

Cascading Style Sheets: Introduction, using Styles, simple examples, your own styles, properties and values in styles, style sheet, formatting blocks of information, layers.

UNIT – III

Introduction to JavaScript: What is DHTML, JavaScript, basics, variables, string manipulations, mathematical functions, statements, operators, arrays, functions. Objects in JavaScript: Data and objects in JavaScript, regular expressions, exception handling

UNIT – IV

DHTML with JavaScript: Data validation, opening a new window, messages and confirmations, the status bar, different frames, rollover buttons, moving images,

UNIT – V

XML: defining data for web applications, basic XML, document type definition, presenting XML, document object model. Web Services
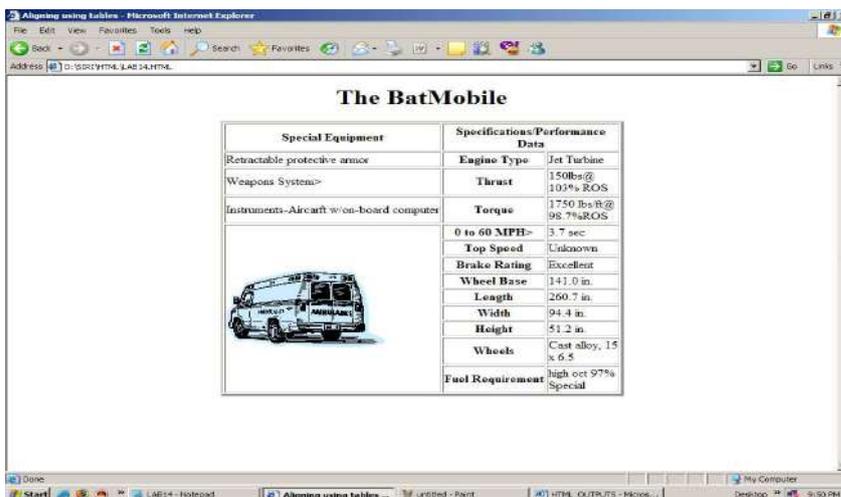

References:

1. Harvey M. Deitel and Paul J. Deitel, **"Internet & World Wide Web How to Program", 4/e,** Pearson Education.

2. Uttam Kumar Roy, Web Technologies from Oxford University Press

Student Activities:
1. **Prepare a web site for your college**
2. **Prepare your personal website**

III YEAR VI SEMESTER
## Paper-VII: Elective-C
Web Technologies Lab

1. Write a HTML program illustrating text formatting.

2. Illustrate font variations in your HTML code.

3. Prepare a sample code to illustrate links between different sections of the page.

4. Create a simple HTML program to illustrate three types of lists.

5. Embed a calendar object in your web page.

6. Create an applet that accepts two numbers and perform all the arithmetic operations on them.

7. Create nested table to store your curriculum.

8. Create a form that accepts the information from the subscriber of a mailing system.
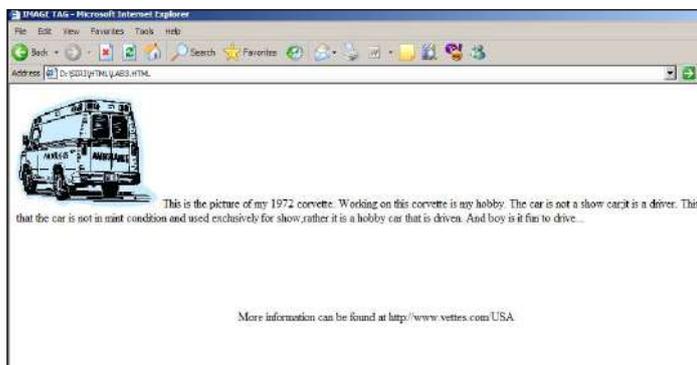
9. Design the page as follows:

11. Using "table" tag, align the images as follows:

12. Divide the web page as follows:
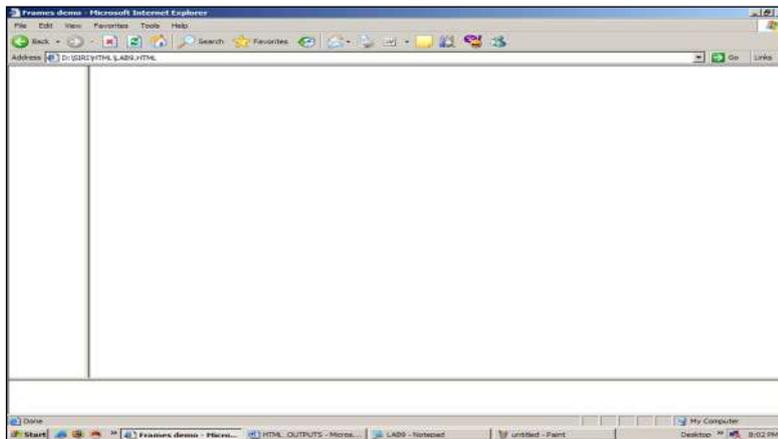
13. Design the page as follows:

14. Illustrate the horizontal rulers in your page.

15. Create a help file as follows:

16. Create a form using form tags(assume the form and fields).
17. Create a webpage containing your biodata(assume the form and fields).
18. Write a html program including style sheets.
20. Write a html program to layers of information in web page.
21. Create a static webpage.

(Cluster 1) Paper-VIII: Elective –A-1
Foundations of Data Science

Course Objectives

Modern scientific, engineering, and business applications are increasingly dependent on data, existing traditional data analysis technologies were not designed for the complexity of the modern world. Data Science has emerged as a new, exciting, and fast-paced discipline that explores novel statistical, algorithmic, and implementation challenges that emerge in processing, storing, and extracting knowledge from Big Data.

Course Outcomes

1. Able to apply fundamental algorithmic ideas to process data.

2. Learn to apply hypotheses and data into actionable predictions.

3. Document and transfer the results and effectively communicate the findings using visualization techniques.

UNIT I

INTRODUCTION TO DATA SCIENCE : Data science process – roles, stages in data science project – working with data from files – working with relational databases – exploring data – managing data – cleaning and sampling for modelling and validation – introduction to NoSQL.

UNIT II

MODELING METHODS : Choosing and evaluating models – mapping problems to machine learning, evaluating clustering models, validating models – cluster analysis – K-means algorithm, Naïve Bayes – Memorization Methods – Linear and logistic regression – unsupervised methods.

UNIT III

INTRODUCTION TO R Language: Reading and getting data into R – ordered and unordered factors – arrays and matrices – lists and data frames – reading data from files.

UNIT IV

PROBABILITY DISTRIBUTIONS in R - Binomial, Poisson, Normal distributions. - Manipulating objects - data distribution.

UNIT V

DELIVERING RESULTS : Documentation and deployment – producing effective presentations– Introduction to graphical analysis – plot() function – displaying multivariate data – matrix plots – multiple plots in one window - exporting graph - using graphics parameters in R Language. Case studies.

Reference Books

1.Nina Zumel, John Mount, "Practical Data Science with R", Manning Publications, 2014.
2.Jure Leskovec, Anand Rajaraman, Jeffrey D.Ullman, "Mining of Massive Datasets",
  Cambridge University Press, 2014.
3.Mark Gardener, "Beginning R - The Statistical Programming Language", John Wiley
  & Sons, Inc., 2012.
4.W. N. Venables, D. M. Smith and the R Core Team, "An Introduction to R", 2013.
5.Tony Ojeda, Sean Patrick Murphy, Benjamin Bengfort, Abhijit Dasgupta, "Practical Data
  Science Cookbook", Packt Publishing Ltd., 2014.
6.Nathan Yau, "Visualize This: The FlowingData Guide to Design, Visualization,
  and Statistics", Wiley, 2011.
7.Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, "Professional Hadoop
  Solutions", Wiley, ISBN: 9788126551071, 2015.

Student Activity:

> 1. **Collect data from any real time system and create clusters using any clustering algorithm**
> 2. **Read the student exam data in R perform statistical analysis on data and print results.**

III YEAR VI SEMESTER
(Cluster 1) Paper-VIII: Elective –A-1
Foundations of Data Science Lab

Objectives :
- R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
- R has an effective data handling and storage facility,
- R provides a suite of operators for calculations on arrays, lists, vectors and matrices.
- R provides a large, coherent and integrated collection of tools for data analysis.

Outcomes:
1) At end student will learn to handle the data through R.
2) Student will familiar with loading and unloading of packages.

I. Installing R and R studio

II. Basic Operations in r

1. Arthematic Operations
2. Comments and spacing
3. Logical Operators - <, <=, >, >=, = , !=, &&, 1

III.
1. Getting data into R, Basic data manipulation
2. Vectors, Materials, operation on vectors and matrices.

IV.
1. Basic Plotting
2. Quantitative data
3. Frequency plots
4. Box plots
5. Scatter plot
6.Categorial data
7. Bar charts
8. Pie charts

V. Loops and functions

1. if, if else, while, for break, next, repeat.
2. Basic functions- Print(), exp( ), Log( ), sqrt( ), abs( ), sin( ), Cos( ), tan( ), factorial( ), rand ().

BIG DATA TECHNOLOGY

Course Objective

The Objective of this course is to provide practical foundation level training that enables immediate and effective participation in big data projects. The course provides grounding in basic and advanced methods to big data technology and tools, including MapReduce and Hadoop and its ecosystem.

Course Outcome

1. Learn tips and tricks for Big Data use cases and solutions.

2. Learn to build and maintain reliable, scalable, distributed systems with Apache Hadoop.

3. Able to apply Hadoop ecosystem components.

UNIT I

INTRODUCTION TO BIG DATA: Introduction – distributed file system – Big Data and its importance, Four V's in bigdata, Drivers for Big data, Big data analytics, Big data applications. Algorithms using map reduce, Matrix-Vector Multiplication by Map Reduce.

UNIT II

INTRODUCTION HADOOP : Big Data – Apache Hadoop & Hadoop EcoSystem – Moving Data in and out of Hadoop – Understanding inputs and outputs of MapReduce - Data Serialization.

UNIT- III

HADOOP ARCHITECTURE: Hadoop Architecture, Hadoop Storage: HDFS, Common Hadoop Shell commands , Anatomy of File Write and Read., NameNode, Secondary NameNode, and DataNode, Hadoop MapReduce paradigm, Map and Reduce tasks, Job, Tasktrackers - Cluster Setup – SSH & Hadoop Configuration – HDFS Administering – Monitoring & Maintenance.

UNIT-IV

HIVE AND HIVEQL, HBASE:-Hive Architecture and Installation, Comparison with Traditional Database, HiveQL - Querying Data - Sorting And Aggregating, Map Reduce Scripts, Joins & Subqueries,

UNIT-V

HBase concepts- Advanced Usage, Schema Design, Advance Indexing - Zookeeper - how it helps in monitoring a cluster, HBase uses Zookeeper and how to Build Applications with Zookeeper.

Reference Books

1. Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, "Professional Hadoop Solutions", Wiley, ISBN: 9788126551071, 2015.

   2 .Big Data Black Book(Covers Hadoop 2, Map Reduce, Hive, Yarn, Pig & Data Visualization)- Dream Tech Publications

3.Chris Eaton, Dirk deroos et al. , "Understanding Big data ", McGraw Hill, 2012.
4. Tom White, "HADOOP: The definitive Guide" , O Reilly 2012.
5. Vignesh Prajapati, "Big Data Analytics with R and Haoop", Packet Publishing 2013.

6. Tom Plunkett, Brian Macdonald et al, "Oracle Big Data Handbook", Oracle Press, 2014.
7. Jy Liebowitz, "Big Data and Business analytics",CRC press, 2013.

Student Activity:

1. Collect real time data and justify how it has become Big Data

2. Reduce the dimensionality of a big data using your own map reducer

III YEAR VI SEMESTER
(Cluster 1) Paper-VIII: Elective –A-2

BIG DATA TECHNOLOGY LAB

Objectives:

- Understand what Hadoop is
- Understand what Big Data is
- Learn about other open source software related to Hadoop

Outcomes:
  i)   Get help on the various Hadoop commands
  ii)   Observe a Map-Reduce job in action

1. Implement the following Data Structures in Java
 a) Linked Lists
 b) Stacks
 c) Queues
 d) Set
 e) Map

2. (i) Perform setting up and Installing Hadoop in its three operating modes: Standalone
     Pseudo distributed
     Fully distributed
   (ii) Use the web based tools to monitor your Hadoop setup.

3. Implement the following file management tasks in Haddop.
   Adding files and directories
   Retrieving files
   Deleting files

Distributed Systems

Course Objectives

To expose the fundamentals of distributed computer systems, assuming the availability of facilities for data transmission.

To discuss multiple levels of distributed algorithms, distributed file systems, distributed databases, security and protection.

Course Outcomes

Create models for distributed systems.

Apply different techniques learned in the distributed system.

UNIT I

Introduction to Distributed Computing Systems, System Models, and Issues in Designing a Distributed Operating System, Examples of distributed systems.

UNIT II

Features of Message Passing System, Synchronization and Buffering, Introduction to RPC and its models, Transparency of RPC, Implementation Mechanism, Stub Generation and RPC Messages, Server Management, Call Semantics, Communication Protocols and Client Server Binding.

UNIT III

Introduction, Design and implementation of DSM system, Granularity and Consistency Model, Advantages of DSM, Clock Synchronization, Event Ordering, Mutual exclusion, Deadlock, Election Algorithms.

UNIT IV

Task Assignment Approach, Load Balancing Approach, Load Sharing Approach, Process Migration and Threads.

UNIT V

File Models, File Accessing Models, File Sharing Semantics, File Caching Schemes, File Replication, Atomic Transactions, Cryptography, Authentication, Access control and Digital Signatures.

Reference Books

1.Pradeep. K. Sinha: " Distributed Operating Systems: Concepts and Design " , PHI, 2007.
2 .George Coulouris, Jean Dollimore, Tim Kindberg: " Distributed Systems" , Concept and Design, 3rd Edition, Pearson Education, 2005.

Student Activity

1. Implementation of Distributed Mutual Exclusion Algorithm.

2. Create a Distributed Simulation Environment.

Distributed Systems Lab

Objective:
It covers all the aspects of distributed system. It introduce its readers to basic concepts of middleware, states of art middleware technology

Outcomes:
1. Students will get the concepts of Inter-process communication
2. Students will get the concepts of Distributed Mutual Exclusion and Distributed Deadlock Detection algorithm.

1. To study client server based program using RPC.

2. To study Client server based program using RMI.

3. To study Implementation of Clock Synchronization (Logical/Psysical)

4. To study Implementation of Election algorithm.

5. To study Implementation of Mutual Exclusion algorithms.

6. To write program multi-threaded client/server processes.

7. To write program to demonstrate process/code migration.

Cloud Computing

Course Objectives:
The student will learn about the cloud environment, building software systems and components that scale to millions of users in modern internet, cloud concepts capabilities across the various cloud service models including Iaas, Paas, Saas, and developing cloud based software applications on top of cloud platforms.

Course Outcomes
1. Compare the strengths and limitations of cloud computing
2. Identify the architecture, infrastructure and delivery models of cloud computing
3. Apply suitable virtualization concept.
4. Choose the appropriate cloud player , Programming Models and approach.
5. Address the core issues of cloud computing such as security, privacy and interoperability
6. Design Cloud Services and Set a private cloud

Unit 1

Cloud Computing Overview – Origins of Cloud computing – Cloud components - Essential characteristics – On-demand self-service , Broad network access , Location independent resource pooling , Rapid elasticity , Measured service

Unit II

Cloud scenarios – Benefits: scalability , simplicity , vendors ,security. Limitations – Sensitive information - Application development – Security concerns - privacy concern with a third party - security level of third party - security benefits Regularity issues: Government policies

Unit III

Cloud architecture: Cloud delivery model – SPI framework , SPI evolution , SPI vs. traditional IT Model
Software as a Service (SaaS): SaaS service providers – Google App Engine, Salesforce.com and google platfrom – Benefits – Operational benefits - Economic benefits – Evaluating SaaS **Platform as a Service** (PaaS): PaaS service providers – Right Scale – Salesforce.com – Rackspace – Force.com – Services and Benefits

Unit IV
Infrastructure as a Service (IaaS): IaaS service providers – Amazon EC2 , GoGrid – Microsoft soft implementation and support – Amazon EC service level agreement – Recent developments – Benefits
Cloud deployment model : Public clouds – Private clouds – Community clouds - Hybrid clouds - Advantages of Cloud computing

Unit V

Virtualization: Virtualization and cloud computing - Need of virtualization – cost , administration , fast deployment , reduce infrastructure cost - limitations

Types of hardware virtualization: Full virtualization - partial virtualization - para virtualization

Desktop virtualization: Software virtualization – Memory virtualization - Storage virtualization – Data virtualization – Network virtualization **Microsoft Implementation**: Microsoft Hyper V – Vmware features and infrastructure – Virtual Box - Thin client

Reference Books

1. Cloud computing a practical approach - Anthony T.Velte , Toby J. Velte Robert Elsenpeter TATA McGraw- Hill , New Delhi - 2010
2. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online - Michael Miller - Que 2008
3. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier.
4. Cloud Computing, A Hands on approach, Arshadeep Bahga, Vijay Madisetti, University Press
5. Mastering Cloud Computing, Foundations and Application Programming, Raj Kumar Buyya, Christenvecctiola, S Tammarai selvi, TMH

Student Activity:
1. Prepare the list of companies providing cloud services category wise.
2. Create a private cloud using local server

III YEAR VI SEMESTER
(Cluster 2) Paper-VIII: Elective –B-2

Cloud Computing Lab

Outcomes: Learner will be able to…
1. Appreciate cloud architecture
2. Create and run virtual machines on open source OS
3. Implement Infrastructure, storage as a Service.

Use Eucalyptus or Open Nebula or equivalent to set up the cloud and demonstrate.
1. Find procedure to run the virtual machine of different configuration. Check how many virtual machines can be utilized at particular time.
2. Find procedure to attach virtual block to the virtual machine and check whether it holds the
data even after the release of the virtual machine.
3. Install a C compiler in the virtual machine and execute a sample program.
4. Show the virtual machine migration based on the certain condition from one node to the other.
5. Find procedure to install storage controller and interact with it.

1. Introduction to cloud computing.
2. Creating a Warehouse Application in Sales Force.com.
3. Creating an Application in Sales Force.com using Apex programming Language.
4. Implementation of SOAP web services in C#/ JAVA Applications.
5. Implementation of Para- Virtualization using VM ware's workstation/ Oracle's Virtual Box and Guest O.S.
6. Case study: PAAS (Face book, Google App Engine)
7. Case Study: Amazon web services.

III YEAR VI SEMESTER
(Cluster C) Paper-VIII: Elective –C-1
Paper-VIII: PHP & MySql, Wordpress

Course Objectives
To introduce the concept of PHP and to give basic Knowledge of PHP. Learn about PHP Syntax., Arrays, PHP Loops, PHP and MySQL connectivity, PHP form validation, PHP form handling. Overview of MySQL and PHPMyAdmin, Understand basic concepts of how a database stores information via tables, Understanding of SQL syntax used with MySQL, Learn how to retrieve and manipulate data from one or more tables, Know how to filter data based upon multiple conditions, Updating and inserting data into existing tables, Learning how the relationships between tables will affect the SQL, The advantages of store procedures with storing data using variables and functions, How SQL can be used with programming languages like PHP to create dynamic websites for visitors, Review of some sample PHP projects interacting with MySQL.

Course Outcomes
After completing this course satisfactorily, a student will be able to:
1. Introduction to web development with PHP
2. How to code a PHP application
3. Introduction to relational databases and MySQL
4. How to use PHP with a MySQL database
5. How to use the MVC pattern to organize your code
6. How to test and debug a PHP application
7. How to work with form data
8. How to code control statements
9. How to work with strings and numbers
10. How to work with dates
11. How to create and use arrays
12. How to work with cookies and sessions
13. How to create and use functions
14. How to use regular expressions, handle exceptions, and validate data

UNIT I
Installing and Configuring MySQL: Current and Future Versions of MySQl, How to Get MySQL, Installing MySQL on Linux, Windows, Trouble Shooting your Installation, Basic Security Guidelines, Introducing MySQL Privilege System, Working with User Privileges. Installing and Configuring Apache: Current and future versions of Apache, Choosing the Appropriate Installation Method, Installing Apache on Linux, Windows, Apache Configuration File Structure, Apache Log Files, Apache Related Commands, Trouble Shooting. Installing and Configuring PHP: Building PHP on Linux with Apache, Windows, php.ini.Basics, The Basics of PHP scripts. The Building blocks of PHP: Variables, Data Types, Operators and Expressions, Constants. Flow Control Functions in PHP: Switching Flow, Loops, Code Blocks and Browser Output.

UNIT II

Working with Functions: What is function?, Calling functions, Defining Functions, Returning the values from User-Defined Functions, Variable Scope, Saving state between Function calls with the static statement, more about arguments. Working with Arrays: What are Arrays?, Creating Arrays, Some Array-Related Functions.
Working with Objects: Creating Objects, Object Instance
Working with Strings, Dates and Time: Formatting strings with PHP, Investigating Strings with PHP, Manipulating Strings with PHP, Using Date and Time Functions in PHP.

UNIT III

Working with Forms: Creating Forms, Accessing Form Input with User defined Arrays, Combining HTML and PHP code on a single Page, Using Hidden Fields to save state, Redirecting the user, Sending Mail on Form Submission, Working with File Uploads. Working with Cookies and User Sessions: Introducing Cookies, Setting a Cookie with PHP, Session Function Overview, Starting a Session, Working with session variables, passing session IDs in the Query String, Destroying Sessions and Unsetting Variables, Using Sessions in an Environment with Registered Users. Working with Files and Directories: Including Files with inclue(), Validating Files, Creating and Deleting Files, Opening a File for Writing, Reading or Appending, Reading from Files, Writing or Appending to a File, Working with Directories, Open Pipes to and from Process Using popen(), Running Commands with exec(), Running Commands with system() or passthru(). Working with Images: Understanding the Image-Creation Process, Necessary Modifications to PHP, Drawing a New Image, Getting Fancy with Pie Charts, Modifying Existing Images, Image Creation from User Input.

UNIT IV

Introduction to MySQL and Interfacing with Databases through PHP
 Understanding the database design process: The Importance of Good Database Design, Types of Table Relationships, Understanding Normalization. Learning basic SQL Commands: Learning the MySQL Data types, Learning the Table Creation Syntax, Using Insert Command, Using SELECT Command, Using WHERE in your Queries, Selecting from Multiple Tables, Using the UPDATE command to modify records, Using RELACE Command, Using the DELETE Command, Frequently used string functions in MySQL, Using Date and Time Functions in MySQL. Using Transaction and stored procedures in MySQL: What is Transaction?, What are Stored Procedures? Interacting with MySQL using PHP: MySQL Versus MySQLi Functions, Connecting to MySQL with PHP, Working with MySQL Data. Creating an Online Address Book: Planning and Creating Database Tables, Creating Menu, Creating Record Addition Mechanism, Viewing Records, Creating the Record Deletion Mechanism, Adding Sub-entities to a Record.

UNIT V

Word press: Introduction to word press, servers like wamp, bitnami e.tc, installing and configuring word press, understanding admin panel, working with posts and pages, using editor, text formatting with shortcuts, working with media-Adding, editing, deleting media elements, working with widgets, menus. Working with themes-parent and child themes, using featured images, configuring settings, user and user roles and profiles, adding external links, extending word press with plug-ins. Customizing the site, changing the appearance of site using css .

REFERENCE BOOKS

1. Julie C. Meloni, PHP MySQL and Apache, SAMS Teach yourself, Pearson Education (2007).
2. Xue Bai Michael Ekedahl, The web warrior guide to Web Programming, Thomson (2006).

Student activity:

1. **Creation of a webpage using wordpress**
2. **Creation of student database of the college**

PHP, MySql & Wordpress LAB

MySQL Lab Cycle
Cycle -1
An Enterprise wishes to maintain the details about his suppliers and other corresponding details. For that he uses the following details.
Suppliers (sid: Integer, sname: string, address: string)
Parts (pid: Integer, pname: string, color: string)
Catalog (sid: integer, pid: integer, cost: real)
The catalog relation lists the prices charged for parts by suppliers.

Write the following queries in SQL:

1. Find the pnames of parts for which there is some supplier.
2. Find the snames of suppliers who supply every part.
3. Find the snames of supplier who supply every red part.
4. Find the pnames of parts supplied by London Supplier abd by no one else.
5. Find the sid's of suppliers who charge more for some part than the average cost of that part.
6. For each part, find the sname of the supplier who charges the most for that part.
7. Find the sid's of suppliers who supply only red parts.
8. Find the sid's of suppliers who supply a red and a green part.
9. Find the sid's of suppliers who supply a red or green part.
10. Find the total amount has to pay for that suppler by part located from London.

Cycle – 2
An organisation wishes to maintain the status about the working hours made by his employees. For that he uses the following tables.

Emp (eid: integer, ename: string, age: integer, salary: real)
Works (eid: integer, did: integer, pct_time: integer)
Dept (did: integer, budget: real, managerid: integer)
An employee can work in more than one department; the pct_time field of the works relation shows the percentage of time that a given employee works in a given department.

Resolve the following queries.
1. Print the names and ages of each employee who works in both Hardware and Software departments.
2. For each department with more than 20 full time equivalent employees (i.e., where the part-time and full-time employees add up to at least that many full-time employees), print the did's together with the number of employees that work in that department.
3. Print the name of each employee whose salary exceeds the budget of all of the departments that he or she work in.
4. Find the managerid's of managers who manage only departments with budgets greater than 1,000,000.
5. Find the enames of managers who manage the departments with largest budget.
6. If a manager manages more than one department, he or she controls the sum of all the budgets for those departments. Find the managerid's of managers who control more than 5,000,000.
7. Find the managerid's of managers who control the highest amount.
8. Find the average manager salary.

PHP Lab Cycle
1. Write a PHP program to Display "Hello"
2. Write a PHP Program to display the today's date.
3. Write a PHP Program to read the employee details.
4. Write a PHP Program to display the
5. Write a PHP program to prepare the student marks list.
6. Write a PHP program to generate the multiplication of two matrices.
7. Write a PHP Application to perform demonstrate the college website.
8. Write a PHP application to add new Rows in a Table.
9. Write a PHP application to modify the Rows in a Table.
10. Write a PHP application to delete the Rows from a Table.
11. Write a PHP application to fetch the Rows in a Table.
12. Develop an PHP application to make following Operations
    i.   Registration of Users.
    ii. Insert the details of the Users.
    iii. Modify the Details.
    iv. Transaction Maintenance.
        a) No of times Logged in
        b) Time Spent on each login.
        c) Restrict the user for three trials only.
        d) Delete the user if he spent more than 100 Hrs of transaction.

Wordpress Lab

1. Installation and configuration of word press.

2. Create a site and add a theme to it.

Paper-VIII: Advanced Java Script
JQUERY /AJAX / JSON / Angular JS

Course Objective:
To impart knowledge in designing a webpage in a structured way by using advanced java script ie., using different scripting languages.

Course Outcomes
On completing the subject, students will be able to: create a dynamic website using advanced features of JavaScript and create a website with good and attractive design

UNIT I
JQuery – Basics: String, Numbers, Boolean, Objects, Arrays, Functions, Arguments, Scope, Built-in Functions. jQuery – Selectors: CSS Element Selector, CSS Element ID Selector, CSS Element Class Selector, CSS Universal Selector, Multiple Elements E, F, G Selector, Callback Functions. jQuery – DOM Attributes: Get Attribute Value, Set Attribute Value. jQuery – DOM Traversing : Find Elements by index, Filtering out Elements, Locating Descendent Elements, JQuery DOM Traversing Methods.

UNIT II
jQuery – CSS Methods : Apply CSS Properties, Apply Multiple CSS Properties, Setting Element Width & Height, JQuery CSS Methods. jQuery – DOM Manipulation Methods: Content Manipulation, DOM Element Replacement, Removing DOM Elements, Inserting DOM elements, DOM Manipulation Methods. jQuery – Events Handling: Binding event handlers, Removing event handlers, Event Types, The Event Object, The Event Attributes. jQuery – Effects: JQuery Effect Methods, jQuery Hide and Show, jQuery Toggle, jQuery Slide – slideDown, slideUp, slideToggle, jQuery Fade – fadeIn, fadeOut, fadeTo, jQuery Custom Animations

UNIT III
Intro to **jQuery UI**, Need of jQuery UI in real web sites, Downloading jQuery UI, Importing jQuery UI, Draggable, Droppable, Resizable, Selectable, Sortable, Accordion, Auto Complete, Button Setϖ, Date Picker, Dialog, Menu, Progress Bar, Slider, Spinner, Tabs, Tooltip, Color Animation, Easing Effects, addClass, removeClass, Effects, jQuery UI themes, Customizing jQuery UI widgets / plug-ins, jQuery UI with CDN, Consuming jQuery Plug-ins from 3rd party web sites jQuery Validations, Intro to jQuery validation plug-in, Using jQuery validation plug-in, Regular expressions.

UNIT IV
Intro to AJAX, Need of AJAX in real web sites, Getting database data using jQuery-AJAX, Inserting, Updating, Deleting database data using jQuery-AJAX Grid Development using jQuery-AJAX

Intro to **JSON** JSON syntax, Need of JSON in real web sites, JSON object, JSON array, Complex JSON objects, Reading JSON objects using jQuery.

UNIT V
Intro to **AngularJS,** Need of AngularJS in real web sites, Downloading AngularJS, AngularJS first example, AngularJS built-in directives, AngularJS expressions,

AngularJS modules, AngularJS controllers, AngularJS scope AngularJS dependency injection AngularJS, bootstrapping AngularJS data bindings, AngularJS $watch, AngularJS filters, AngularJS events, AngularJS AJAX, Ng-repeat, AngularJS with json arrays, AngularJS registration form and login form, AngularJS CRUD operations, AngularJS Animations, AngularJS validationsϖ AngularJS $q, AngularJS custom values, AngularJS custom factories, AngularJS custom services, AngularJS custom directives, AngularJS custom providers, AngularJS Routing, AngularUI Routing.

Reference Books

1. jQuery UI 1.8: The User Interface Library for jQuery by Dan Wellman
2. jQuery Fundamentals by Rebecca Murphey
3. Ajax: The Complete Reference by Thomas A. Powell
4. Pro AngularJS by Adam Freeman Kindle Edition

Student Activity:

1. Creation of website for a small scale company
2. Creation of website for a student database

III YEAR V SEMESTER

Advanced Java Script
JQUERY /AJAX / JSON / Angular JS

1. Using jQuery find all textareas, and makes a border. Then adds all paragraphs to the jQuery object to set their borders red.

2. Using jQuery add the class "w3r_font_color" and w3r_background to the last paragraph element.

3. Using jQuery add a new class to an element that already has a class.

4. Using jQuery insert some HTML after all paragraphs.

5. Using jQuery insert a DOM element after all paragraphs.

6. Convert three headers and content panels into an accordion. Initialize the accordion and specify the animate option

7. Convert three headers and content panels into an accordion. Initialize the accordion and specify the height.

8. Create a pre-populated list of values and delay in milliseconds between a keystroke occurs and a search is performed.

9. Initialize the button and specify the disable option.

10. Initialize the button and specify an icon on the button.

11. Initialize the button and do not show the label.

12. Create a simple jQuery UI Date picker. Now pick a date and store it in a textbox.

13. Initialize the date picker and specify a text to display for the week of the year column heading.

PROJECT-2

Follow SDLC process for real time applications and develop real time application project

The objective of the project is to motivate them to work in emerging/latest technologies, help the students to develop ability, to apply theoretical and practical tools/techniques to solve real life problems related to industry, academic institutions and research laboratories.

The project is of 5 hours/week for one (semester VI) semester duration and a student is expected to do planning, analyzing, designing, coding, and implementing the project. The initiation of project should be with the project proposal. The synopsis approval will be given by the project guides.

The project proposal should include the following:

Title

Objectives

Input and output

Details of modules and process logic

Limitations of the project

Tools/platforms, Languages to be used

Scope of future application

The Project work should be either an individual one or a group of not more than three members and submit a project report at the end of the semester. The students shall defend their dissertation in front of experts during viva-voce examinations.