Shiv Chhatrapati Shikshan Sanstha's

Rajarshi Shahu Mahavidyalaya, Latur (Autonomous)

Curriculum

2021-2022

B.Sc. C.S.

(CC/AECC/SEC/GE)

UG Third Year Semester V & VI

Under CBCS

Three Year Degree Programme in B.Sc.C.S.

(Six Semester Course)

Syllabi approved by the Board of Studies in

B.Sc.C.S. with effect from June, 2019

Rajarshi Shahu Mahavidyalaya (Autonomous), Latur
Department of Information Technology

Syllabus outline of B.Sc.C.S. Third Year
Under CBCS Pattern

Semester	Course Code	Course Title	Internal		End Semester		Total	
			Credit	Marks	Credit	Marks	Credit	Marks
Semester – V	**1. Skill Enhancement Course**							
	U-APR-601	Aptitude & Reasoning	--	20	--	30	2	50
	2. Discipline Specific Elective Course (Choose any four DSEC papers)							
	U-DIP-602	Digital Image Processing	--	20	--	30	3	50
	U-RDB-603	RDBMS through Oracle	--	20	--	30	3	50
	U-ASP-604	Web Designing Using ASP.Net	--	20	--	30	3	50
	U-PWP-605	Programming with Python	--	20	--	30	3	50
	U-INP-606	Internet Programming	--	20	--	30	3	50
	U-COG-607	Computer Graphics	--	20	--	30	3	50
	Env. Studies	(NCBC)						
	3. Practical / Lab Course							
	U-LAC-608	Lab Course- XVII	--	20	--	30	2	50
	U-LAC-609	Lab Course-XVIII	--	20	--	30	2	50
	U-LAC-610	Lab Course-XIX	--	20	--	30	2	50
	U-LAC-611	Lab Course-XX	--	20	--	30	2	50
	U-LAC-612	Lab Course-XXI	--	20	--	30	2	50
	U-LAC-613	Lab Course-XXII	--	20	--	30	2	50
	U-PRW-542	Mini Project / Internship	--	20	--	30	2	50
	U-SEM-543	Seminar	--	20	--	30	2	50
	Total (V)						**26**	**550**

Semester	Course Code	Course Title	Credits / Marks				Total	
			Internal		End Semester			
			Credit	Marks	Credit	Marks	Credit	Marks
Semester – VI	**1. Skill Enhancement Course**							
	U-PDI-701	Personality Development & Interview Techniques	--	20	--	30	2	50
	2. Discipline Specific Elective Course (Choose any four DSEC papers)							
	U-PCD-702	Principles of Compiler Design	--	20	--	30	3	50
	U-WPP-703	Web Programming Using PHP	--	20	--	30	3	50
	U-DBA-704	Database Administration	--	20	--	30	3	50
	U-ADA-705	Advanced Data Structures and Algorithms	--	20	--	30	3	50
	U-DAM-706	Data Mining	--	20	--	30	3	50
	U-CLC-707	Cloud Computing	--	20	--	30	3	50
	3. Practical / Lab Course							
	U-LAC-708	Lab Course-XXIII	--	20	--	30	2	50
	U-LAC-709	Lab Course -XXIV	--	20	--	30	2	50
	U-LAC-710	Lab Course- XXV	--	20	--	30	2	50
	U-LAC-711	Lab Course-XXVI	--	20	--	30	2	50
	U-LAC-712	Lab Course-XXVII	--	20	--	30	2	50
	U-LAC-713	Lab Course- XXVIII	--	20	--	30	2	50
	U-PRW-641	Major Project	--	40	--	60	4	100
	Total (VI)						**26**	**550**
	Total (V + VI)						**52**	**1100**
	Total (I + II + III + IV+V+VI)						**144**	**3000**

Semester - V

<div align="center">

Course Title: Aptitude and Reasoning
Course Code: U-APR-601

</div>

Total Lectures: 50   **Total Marks: 50**
                                                          Credits: 02

Learning Objectives
This course is designed to suit the need of the outgoing students and to acquaint them with frequently asked patterns in quantitative aptitude and logical reasoning during various examinations and campus interviews.

Course Outcome
On successful completion of the course the students will be able to:
- Understand the basic concepts of quantitative ability
- Understand the basic concepts of logical reasoning Skills
- Acquire satisfactory competency in use of reasoning
- Solve campus placements aptitude papers covering Quantitative Ability, Logical Reasoning Ability
- Compete in various competitive exams like CAT, CMAT, GATE, GRE, GATE, UPSC, GPSC etc.

<div align="center">

Syllabus

</div>

UNIT - I
1. Quantitative Ability (Basic Mathematics)
1.1.   Number Systems
1.2.   LCM and HCF
1.3.   Decimal Fractions
1.4.   Simplification
1.5.   Square Roots and Cube Roots
1.6.   Average
1.7.   Problems on Ages
1.8.   Surds & Indices
1.9.   Percentages
1.10   Problems on Numbers

UNIT – II
 2.   Quantitative Ability (Applied & Engineering Mathematics)
2.1.   Logarithm
2.2.   Permutation and Combinations
2.3.   Probability
2.4.   Profit and Loss
2.5.   Simple and Compound Interest
2.6.   Time, Speed and Distance
2.7.   Time & Work
2.8.   Ratio and Proportion
2.9.   Area
2.10   Mixtures and Allegation

UNIT – III
2. Data Interpretation
3.1. Data Interpretation
3.2. Tables
3.3. Column Graphs
3.4. Bar Graphs
3.5. Line Charts
3.6. Pie Chart
3.7. Venn Diagrams

UNIT – IV
4. Logical Reasoning (Deductive Reasoning)
4.1. Analogy
4.2. Blood Relation
4.3   Directional Sense
4.4. Number and Letter Series
4.5. Coding – Decoding
4.6. Calendars
4.7. Clocks
4.8. Venn Diagrams
4.9. Seating Arrangement
4.10. Syllogism
4.11. Mathematical Operations

Reference books:

1. A Modern Approach To Verbal & Non Verbal Reasoning By R S Agarwal
2. Analytical and Logical reasoning By Sijwali B S
3. Quantitative aptitude for Competitive examination By R S Agarwal
4. Analytical and Logical reasoning for CAT and other management entrance test By Sijwali B S
5. Quantitative Aptitude by Competitive Examinations by Abhijit Guha 4 th edition
6. https://prepinsta.com/
7. https://www.indiabix.com/
8. https://www.javatpoint.com/

Course Title: Digital Image Processing
Course Code: U-DIP-602

Total Lectures: 50
Total Marks: 50
Credits: 3

Learning Objectives
 This course will enable a student to:
- Understand image processing concepts
- Study mathematics and algorithms for image processing
- Study various applications in image processing

Course Outcome
After completing this course, student will be able to:
- Describe fundamental steps used in digital image processing
- Understand applications of digital image processing
- Design and implement image processing programs to solve image processing problems
- Describe how digital images are represented and stored efficiently depending on the desired quality, color depth
- Use the mathematical principles of digital image enhancement (contrast, noise)
- Describe and apply the concepts of filtering techniques
- Analyze the algorithms used in image processing to perform designated tasks using MATLAB

Syllabus

UNIT- I   Introduction to DIP
What is digital image processing? Example fields of digital image processing, Fundamental steps in digital image processing, Components of image processing system. Elements of visual perception, Lights and electromagnetic spectrum, Image sensing and acquisition, Image sampling and quantization, Some basic relationship between pixels.

Unit –II   Digital Image Representation using Matlab
Digital Image Representation: Coordinate Conventions, Images as Matrices Reading Images, Displaying Images, Writing Images, Data Classes, Image Types: Intensity Images, Binary Images, Converting between Data Classes and Image Types: Converting between Data Classes, Converting between Image Classes and Types, Array Indexing: Vector Indexing, Matrix Indexing, Selecting Array Dimensions, Some Important Standard Arrays. Introduction to M-Function Programming: M-Files, Operators, Flow Control, Code Optimization, Interactive I/O

Unit- III   Intensity transformation using Matlab
 Intensity Transformation Functions: Function imadjust, Logarithmic and Contrast-Stretching Transformations, Some Utility M-Functions for Intensity Transformations Histogram Processing and Function Plotting: Generating and Plotting Image Histograms, Histogram Equalization, Histogram Matching (Specification) Spatial Filtering: Linear Spatial Filtering, Nonlinear Spatial Filtering

Unit –IV   Frequency Domain Processing and Histogram Processing

Frequency Domain Processing: The 2-D Discrete Fourier Transform, Computing and Visualizing the 2-D DFT in MATLAB, Filtering in the Frequency Domain: Fundamental Concepts, Basic Steps in DFT Filtering, A Model of the Image Degradation/Restoration Process, Color Image Representation in MATLAB: RGB Images, Indexed Images IPT Functions for Manipulating RGB and Indexed Images.

Reference Books:

1. Digital Image Processing Using MATLAB by Rafael C. Gonzalez, Richard E. Woods, Steven L.Eddins
2. Digital Image Processing by Rafael C. Gonzalez, Richard E. Woods

Course Title: RDBMS through Oracle
Course Code: U-RDB-603

Total Lectures: 50   **Total Marks: 50**
                                                                                              Credits: 3

Learning Objectives
This course will enable a student to:
- Explain database concepts, particularly the concepts of relational databases
- Design and implement SQL databases
- Understand and use the Structured Query Language - DDL, DML and DCL.
- Write simple stored procedures and triggers using PL/SQL
- Use and develop application databases.

Course Outcome
After completing this course, student will be able to:
- Understand the role of database in organizations.
- Normalize database tables in order to design and implement correct database systems.
- Understand and use the Structured Query Language (SQL) in depth and obtain ample hands-on practice.
- Understand and use database transactions and concurrency
- Create a Term Project that covers all aspects of designing a database and the SQL requests that run against that database.

.

Syllabus

Unit- I

Chap 1. SQL Statements & Working with tables
    1.1. DDL, DML, DQL, DCL
    1.2. Data types in SQL
    1.3. Creating & Managing Tables
    1.4 Data Constraints
    1.5 SELECT statement with where, Order by and Distinct clause
    1.6 Oracle operators

Chap 2. Advance SQL
    2.1 Oracle Views: Types, creating and managing views
    2.2 SQL Functions: Single Row Functions, Character Functions, Number Functions, Date Functions, Conversion Functions
    2.3 Group by and having clause

Unit –II
Chap 3. Joining Tables &Working with Sub queries
    3.1. What is Join?
    3.2. Natural Join/Inner Join/Equijoin/outer join / self join
    3.3. Joining With 'USING' Clause

3.4. Joining With 'ON' Clause

3.5. Subqueries: Single Row Sub query, Multiple Row Sub query

Chap 4. PL/SQL Overview

4.1. PL/SQL block structure

4.2. Condition logic

4.3 Loops

UNIT- III

Chap 5. Advance in PL/SQL

5.1. Triggers: Triggers Syntax, Types of triggers: Row Level, Statement Level, Before, After, Enabling and Disabling Triggers, Replacing and Dropping Triggers

5.2 Oracle transactions: commit, rollback and savepoint

5.3. Working With Cursor: Types of cursor: Implicit & Explicit cursor, general cursor attributes, use of % TYPE and % ROWTYPE Variable, cursor for loops

UNIT- IV

Chap 6. Concurrency control in Oracle and Database objects

6.1 Locks, types of locks, levels of locks

6.2 Exception Handling in PL/SQL

6.3. PL/SQL Procedures and functions

Reference Books

1. Oracle Database 10g SQL (Osborne ORACLE Press Series) by Jason price, McGrawHill, 0-07-222981-0.
2. Oracle Database 10g PL/SQL Programming by Scott Urman, Ron HARDMAN, MichaleMc Laughlin, Oracle Press, TMH, ISBN-0-07- 059779-0.
3. Oracle Database 10g The Complete Reference By Kevin Loney, Bob Bryla Oracle Press (TATA McGraw Hill Edition) ISBN-13:978-0-07- 059425-8, ISBN-10: 0-07-059425-2

Course Title: Web Designing Using ASP.Net
Course Code: U-ASP-604

Total Lectures: 50   **Total Marks: 50**
                                                          Credit: 3

Learning Objectives
This course will enable a student to:
- Set up a programming environment for ASP.net programs.
- Create & configure an asp.net application using standard .net controls.
- Develop a data driven web application.
- Connect to data sources and managing them.
- Maintain session and controls related information for user used in multiuser web applications.
- Understand the fundamentals of developing modular application by using object oriented methodologies

Course Outcome
After completion of this course, students will be able to:
- Design web applications using ASP.NET
- Use Server Controls in Real-time application
- Understand Concept of Master Page Implementation on Website
- Implement Session in web application
- Build Simple Website

SYLLABUS

UNIT I   Introduction To ASP.NET AND SERVER CONTROLS

1. Introduction to ASP.NET
   a. What is asp.net?
   b. Introduction to visual studio.
   c. Web Page Life Cycle.
   d. IIS
   e. Web.config, global.asax

2. Html and Server Controls and Rich Controls
   a. Html Controls-(Label, TextBox, Button, CheckBox, RadioButton)
   b. Server controls-(Label,TextBox, Button, CheckBox, RadioButton)
   c. Rich controls-(DropDownList, Calender)

UNIT II NAVIGATION AND VALIDATION CONTROL

3. Navigation Control and Master Page
a. Menu Control
b. Navigation functions
c. Master Page
d. Example using Master Page

4. Validation controls
a. Client side validation
b. Serverside validation controls :Compare validator, Required Field Validator, Range Validator, Custom validator
c. Regular Expressions

UNIT-III STATE MANAGEMENT AND DATA CONTROL

5. State Management
a. Session
b. Cookies
c. View State
d. Query String
e. Difference between session cookies and view state

6. Data Controls
a. The GridView : Automatically Generating Columns, Defining Columns
b. Formatting the Grid View-editing and updating data in Grid View
c. Sorting and Paging the GridView : Sorting , Paging
d. Export Grid view Data to Excel or pdf

UNIT-IV WEB SERVICES AND ADO.NET

7. ADO.NET
a. What is ADO.NET?
b. SqlConnection , SqlCommand
c. DataSet, DataAdapter
d. Simple Application using ADO.NET

8. Introduction to Caching and Web Services
Understanding Caching, Introduction to WCF, WPF

Reference Books:
1. Asp.Net Black Book-Steven Holzner
2. Web Publishing- Monica D'Souza
3. Mastering ASP.Net BPB Publication
4. ASP.Net – The Complete Reference Tata McGraw Hill
5. Professional ASP.NET – Wrox Publication

Total Lectures: 55   **Total Marks: 50**
                                                                                        Credits: 3

Learning Objectives
 This course will enable a student to:
1. Explain basic principles of Python programming language
2. Implement object oriented concepts
3. Understanding of scripting and the contributions of scripting languages.
4. Understanding of the built□in objects of Python

Course outcomes
After completion of this course, student will able to:
1. Adapt and combine standard algorithms to solve a given problem (includes numerical as well as non-numerical algorithms)
2. Adequately use standard programming constructs: repetition, selection, functions, composition, modules, aggregated data
3. Identify and repair coding errors in a program
4. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Unit – I   Getting started with python

Introduction to python, features, program output, program input and raw_input(),comments in python, operators, Code blocks and indentation.

Unit –II   Python Basics, Syntax and Style

Variable assignment and Identifiers, Numbers: Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions, Strings: Strings and Operators, String-only Operators, Built-in Functions, String Built-in Methods, Special Features of Strings.
Lists: Operators, Built-in Functions, List Type, Built-in Methods, Special Features of Lists.
Tuples: Tuple Operators and Built-in Functions, Special Features of Tuples

Unit –III   Control Constructs and Exceptions

Conditionals and Loops: if statement, else Statement, elif Statement, while Statement, for Statement, break Statement, continue Statement, pass Statement, else Statement

Errors And Exceptions: What Are Exceptions?, Exceptions in Python, Detecting and Handling Exceptions.

Unit –IV   Functions

What Are Functions?, Calling Functions, Creating Functions, Passing Functions, Formal Arguments, Positional Arguments, Default Arguments, Why Default Arguments?, Default Function Object Argument Example, Variable-length Arguments, Non-keyword Variable Arguments (Tuple) , Keyword Variable Arguments (Dictionary)

Unit –V   Classes and OOP

Introduction, Object-oriented Programming, Classes, Class Attributes, Instances, Instance Attributes, Binding and Method Invocation, Composition, Sub-classing and Derivation Inheritance, Built-in Functions for Classes, Instances, and Other Objects, Type vs. Classes/Instances

Reference Books

1. Core Python Programming, Second Edition By Wesley J. Chun., Publisher: Prentice Hall
2. Learning Python, 5th edition By Mark Lutz, Publisher: O'Reilly

Total Lectures: 55   **Total Marks: 50**
                                                                              Credits: 3

Learning Objectives:
The course has been designed to provide the knowledge of Cascading Style Sheet, Java Script and XML for design of the web page / site.

Course Outcomes:
At the end of this course the successful student will be able to:
• Apply a structured approach to identifying needs, interests, and functionality of a website.
• Design dynamic websites that meet specified needs and interests.
• Write well-structured, easily maintained, standards-compliant CSS code to present HTML pages in different ways.
• Use JavaScript to add dynamic content to pages.
• Select appropriate HTML, CSS, and JavaScript code from public repositories of open source and free scripts that enhances the experience of site visitors.
• Modify existing HTML, CSS, and JavaScript code to extend and alter its functionality, and to correct errors and cases of poor practice.
• Effectively debug JavaScript code, making use of good practice and debugging tools.
• Use JavaScript libraries (e.g. JQuery) to create dynamic pages.

Syllabus

Unit I: HTML Basics and UI Designing
Structure of HTML document, Basic Markup Tags, Working with Table, Working with Images, Creating and Handling Frames. Forms: What is Form?, Form Tag, Method, Action, Input Tag, Type Attribute: Check box, Hidden, Image, Radio, Reset, Submit, Text, other attributes

Unit II: Cascading Style Sheet
Evolution of CSS, Understanding the CSS Syntax, Exploring CSS Selectors (universal, type, class, id, child, descendant, adjacent sibling), Inserting CSS in an HTML document: The Internal Style Sheet The External Style Sheet The Inline Style Sheet Defining Inheritance in CSS Backgrounds and Color Gradients, Fonts and Text Styles, Creating Boxes and Columns, Displaying, Positioning, Floating an Element, List Styles, Table Layouts Pseudo-classes and Pseudo-element

Unit III: JavaScript

Introduction, Client-Side JavaScript, Server-Side JavaScript, JavaScript Objects, JavaScript Security

Operators: Assignment Operators, Comparison Operators, Arithmetic Operators, % (Modulus), ++(Increment), --(Decrement), -(Unary Negation), Logical Operators, Short-Circuit Evaluation, String Operators, Special Operators, ?: (Conditional operator), , (Comma operator), delete, new, this, void

Statements: Break, comment, continue, delete, do...while, export, for, for...in, function, if...else, import, labeled, return, switch, var, while

Core JavaScript (Properties and Methods of Each) : Array, Boolean, Date, Function, Math, Number, Object, String, regExp

Document and its associated objects: document, Link, Area, Anchor, Image, Applet, Layer.

Fundamental JavaScript Directives: In-Line JavaScript, Linking Web Pages to External JavaScript Files, JavaScript Using <script> Tag and attributes,

Event Handling: Events and Event handlers, General information about events, Defining event handlers,

Events:onAbort,onBlur,onChange,onClick,onDblClick,onDragDrop,onError,onFocus,onKeyD own,onKeyPress,onKeyUp,onLoad,onMouseDown,onMouseMove,onMouseOut, onMouseOver,onMoouseUp,onMove,onReset,onResize,onSelect,onSubmit,onUnLoad

Unit IV: Jquery and XML

Fundamentals of jQuery, Loading and using jQuery, jQuery Syntax, jQuery Selectors, Element properties and attributes, Methods to access HTML Attributes, Methods for Traversing, jQuery Events, CSS using jQuery. Introduction to XML: Anatomy of an XML document Creating XML Documents, Creating XML DTDs, XML Schemas, XSL.

Reference Books

1) HTML Complete Reference
2) JavaScript 2.0: The Complete Reference, Tata McGraw Hill
3) XML Complete Reference

Recommended Readings

1) http://www.w3schools.com/css
2) http://jigsaw.w3.org/css-validator

Course Title: Computer Graphics
Course Code: U-COG-607

Total Lectures: 50   **Total Marks: 50**
                                                                Credits: 3

Learning Objectives
This course will enable a student to:
- Understand computer graphics concepts
- Explain co-ordinate systems and different types of printers
- Explain line drawing and clipping algorithms
- Implementation of graphics programs
- Understand and explain 2D and 3D transformations

Course Outcomes
After completing this course, student will be able to:
- Detailed knowledge of the graphics devices
- Understand the theory of 2D and 3D transformations, projection and viewing
- Create graphics designs
- Detailed knowledge of clipping and windowing algorithms.
- Ability to understand, design and implement scene graphs
- Practical skills in graphics programming including scene graph

Syllabus

Unit- I
1. Introduction to Computer Graphics
1.1 Introduction
1.2 Advantages of computer graphics
1.3 Application of computer graphics
1.4 Co-ordinate system: Cartesian and Polar
1.5 Display devices: Cathode Ray Tubes, Color CRT monitors
1.6 Direct View Storage Tube, Plotter, Light pen, Joystick

2. Raster Scan Graphics
2.1 Line segment and line drawing algorithm
2.2 Digital differential Algorithm
2.3 Bresenham's line algorithm

Unit – II
3. Transformation
3.1 Two dimensional transformation
3.2 Matrix representation
3.3 Transformation: Translation, Rotation, Scaling, Reflection, Shear
4. Segmented Display Files
4.1 Segment table
4.2 Functions for segmenting display file
4.3 Posting & unposting segments

4.4 Segment naming scheme
4.5 Default error conditions
4.6 Appending to segments

Unit – III
5 . Clipping & Windowing
5.1 Viewing transformation
5.2 2-D clipping
5.3 End point codes
5.4 Midpoint subdivision algorithm,
5.5 Polygon clipping algorithm (Sutherland-Hodgman algorithm)
5.6 Windowing transformation

6 Display File Compilations
6.1 Display file compiler
6.2 Refresh concurrent with reconstruction
6.3 Free storage allocation
6.4 Display file structure

Unit -IV
7 Geometric Models
7.1 Simple modeling example
7.2 Geometric modeling
7.3 Symbols & instances
7.4 Implementation of Instance transformation

8 Simple Graphics Package

8.1 Ground rules for graphics s/w design
8.2 Function domains
8.3 Graphics primitives
8.4 Windowing function

Reference Books:
1. Principle of Interactive Computer Graphics -Willam Newman& Robert Sproull (TMH)
2. Procedural Elements for Computer Graphics -David F. Rogers (TMH)
3. Computer graphics -A.P.Godse

Course Title: Lab Course XVII (Digital Image Processing)
Course Code: U-LAC-608

Total Marks: 50   **Credits: 2**

_____

Learning Objectives
This course will enable a student to:
- Understand image processing concepts
- Study mathematics and algorithms for image processing
- Study various applications in image processing

Course Outcomes
After completing this course, student will be able to:
- Describe fundamental steps used in digital image processing
- Understand applications of digital image processing
- Design and implement image processing programs to solve image processing problems
- Describe how digital images are represented and stored efficiently depending on the desired quality, color depth
- Use the mathematical principles of digital image enhancement (contrast, noise)
- Describe and apply the concepts of filtering techniques
- Analyze the algorithms used in image processing to perform designated tasks using MATLAB

Proposed Practical List:

1. Study of MATLAB environment

2. Study of reading, writing and showing images

3. Study of image type conversions

4. Working on vectors and matrices

5. Study of M-file and control statements

6. Study of Histogram processing and equalization

7. Study of intensity transformation

8. Study of spatial filtering techniques

9. Study of Image restoration process

10. Program for color image processing

Course Title: Lab Course XVIII (RDBMS through Oracle)
Course Code: U-LAC-609

Total Marks: 50   **Credits: 2**

Learning Objectives
This course will enable a student to:
- Explain database concepts, particularly the concepts of relational databases
- Design and implement SQL databases
- Understand and use the Structured Query Language - DDL, DML and DCL.
- Write simple stored procedures and triggers using PL/SQL
- Use and develop application databases.

Course Outcomes
After completing this course, student will be able to:
- Understand the role of database in organizations.
- Normalize database tables in order to design and implement correct database systems.
- Understand and use the Structured Query Language (SQL) in depth and obtain ample hands-on practice.
- Understand and use database transactions and concurrency
- Create a Term Project that covers all aspects of designing a database and the SQL requests that run against that database

Proposed Practical List
1. Study of DDL statements
2. Study of DML statements
3. Study of DCL statements
4. Study of select statements with different operators. (arithmetic operators ,logicaloperators, range searching, pattern matching, column alias)
5. Study of data constraints
6. Study of oracle views
7. Study of oracle functions
8. Study of group by and having clause
9. Study of Oracle join
10. Study of subqueries
11. Introduction of PL/SQL.
12. PL/SQL Looping & conditional structures
13. Study of Oracle trigger
14. Study of Oracle cursor
15. Study of Oracle stored procedures & functions

<div align="center">**Course Title: Lab Course XIX (ASP.Net)**
Course Code: U-LAC-610</div>

Total Marks: 50   **Credits: 2**

Learning Objectives
This course will enable a student to:
- Set up a programming environment for ASP.net programs.
- Creating & configure an asp.net application using standard .net controls.
- Develop a data driven web application.
- Connecting to data sources and managing them.
- Maintain session and controls related information for user used in multiuser web applications.
- Understand the fundamentals of developing modular application by using object oriented methodologies
-

Course Outcomes
After completion of this course, students will be able to:
- Design web applications using ASP.NET
- Use Server Controls Real-time
- Understand Concept Of Master Page Implementation on Website
- Implement Session in web Application
- Build Simple Website

Practical List
1. Program to Demonstrate Asp.net Web Page Life Cycle
2. Program to Demonstrate Textbox Control
3. Program to Demonstrate Label and Button Controls
4. Program to Demonstrate Dropdown List Control
5. Program to Demonstrate Calendar Control
6. Program to Demonstrate Menu Control
7. Program to Demonstrate Navigation Function
8. Program to Demonstrate Master Page
9. Program to Demonstrate Compare validator, Required Field Validator, Range Validator
10. Program to Demonstrate Custom Validator Control
11. Program to Demonstrate Session
12. Program to Demonstrate Cookies
13. Program to Demonstrate View State
14. Program to Demonstrate Query String
15. Program to Demonstrate Formatting the Grid View
16. Program to Demonstrate Sorting and Paging the Grid View
17. Program to Demonstrate Export Grid view Data to pdf
18. Program to Demonstrate SqlConnection
19. Application using ADO.NET(With SqlServer)
20. Program to Demonstrate DataSet, DataAdapter

Total Marks: 50   **Credits: 2**

Learning Objectives
This course will enable a student to:
1. Explain basic principles of Python programming language
2. Implement object oriented concepts
3. Understanding of scripting and the contributions of scripting languages.
4. Understanding of the built☐in objects of Python

Course outcomes
After completion of this course, student will able to:
1. Adapt and combine standard algorithms to solve a given problem (includes numerical as well as non-numerical algorithms)
2. Adequately use standard programming constructs: repetition, selection, functions, composition, modules, aggregated data
3. Identify and repair coding errors in a program
   Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

Proposed practical list
1. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.
2. Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.
3. Write a program to generate the Fibonacci series.
4. Write a function that reverses the user defined value.
5. Write a function to check the input value is Armstrong and also write the function for Palindrome.
6. Write a recursive function to print the factorial for a given number.
7. Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.
8. Define a function that computes the length of a given list or string.
9. Write a program that takes two lists and returns True if they have at least one common member.
10. Write a Python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.
11. Write a Python program to clone or copy a list
12. Write a Python script to sort (ascending and descending) a dictionary by value.
13. Write a Python script to concatenate following dictionaries to create a new one.
14. Write a Python program to sum all the items in a dictionary.
15. Design a class that store the information of student and display the same

Total Marks: 50   **Credits: 2**

Learning Objectives

   This course has been designed to provide the knowledge of Cascading Style Sheet, Java Script and XML for design of the web page / site.

Course Outcomes

At the end of this course the successful student will be able to:

• Apply a structured approach to identifying needs, interests, and functionality of a website.

• Design dynamic websites that meet specified needs and interests.

• Write well-structured, easily maintained, standards-compliant CSS code to present HTML pages in different ways.

• Use JavaScript to add dynamic content to pages.

• Select appropriate HTML, CSS, and JavaScript code from public repositories of opensource and free scripts that enhances the experience of site visitors.

• Modify existing HTML, CSS, and JavaScript code to extend and alter its functionality, and to correct errors and cases of poor practice.

• Effectively debug JavaScript code, making use of good practice and debugging tools.

• Use JavaScript libraries (e.g. JQuery) to create dynamic pages.

Proposed practical list

1. Design a web page using different text formatting tags.
2. Design a web page with links to different pages and allow navigation between web pages.
3. Design a web page with different tables.
4. Design a webpage using table so that the content appears well placed.
5. Design a web page with a form that uses all types of controls.
6. Creating and Using Internal Style Sheet
7. Creating and Using External cascading style sheet
8. Using JavaScript design, a web page that prints factorial/Fibonacci series/any given series.
9. Design a form and validate all the controls placed on the form using Java Script.
10. Write a JavaScript program to display all the prime numbers between 1 and 100.
11. Write a program to accept a number from the user and display the sum of its digits.
12. Write a java script program to design simple calculator.
13. Design a web page demonstrating different conditional and looping statements.

Course Title: Lab Course XXII (Computer Graphics)
Course Code: U-LAC-613

Total Marks: 50   **Credits: 2**

Learning Objectives

This course will enable a student to:
- Understand computer graphics concepts
- Explain co-ordinate systems and different types of printers
- Explain line drawing and clipping algorithms
- Implementation of graphics programs
- Understand and explain 2D and 3D transformations

Course Outcomes

After completing this course, student will be able to:
- Detailed knowledge of the graphics devices
- Understand the theory of 2D and 3D transformations, projection and viewing
- Create graphics designs
- Detailed knowledge of clipping and windowing algorithms.
- Ability to understand, design and implement scene graphs
- Practical skills in graphics programming including scene graph

Proposed Practical List

1. Program to find the resolution of screen
2. Program to implement DDA
3. Program to implement Bresenham's integer algorithm
4. Program to implement Bresenham's general algorithm
5. Program to demonstrate translation
6. Program to demonstrate rotation
7. Program to demonstrate scaling
8. Program to demonstrate circle and rectangle, ellipse.
9. Program to demonstrate bar and 3dbar
10. Program to draw a national flag
11. Program for creating simple animations.
12 Program for demonstration of setting font style, font name and size.
13. Program for demonstration of getimage and putimage function.
14. Program for demonstration of setfillstyle and floodfill functions.
15. Program to draw multiple shapes using loops.

<div align="center">

Course Title: Mini Project
Course Code: U-PRW-542

</div>

Total Marks: 50   **Credits: 2**

Learning objectives

The student will be able to:

- Discover potential research areas in the field of IT
- Identify real world problems and challenges that need IT based solutions
- Demonstrate an ability to work in teams
- Improve the team building, communication and management skills of the students.
- Conduct a survey of several available literature in the preferred field of study
- Use all concepts of IT in creating a solution for a problem
- Formulate and propose a plan for creating a solution for the research plan identified
- Report and present the findings of the study conducted in the preferred domain

Course outcomes

After completing this course, the students will be able to have hands-on to carry out the Project Work using various platforms and project tools which will provide practical experience. Also,

- Implement what is learnt during course
- Get real-life experience by working in real systems
- Understand the problems faced during project implementation.
- Enhance the problem solving ability by solving the real-time problems.
- Learn team work and appreciate role of each of the team members.

Mini project Description:

• A team of maximum 3 students can be formed for this project and work together to learn working collaboratively.

• The team can choose the project area from the following:

- o Java based project
- o .Net based project
- o A project consisting analysis of an existing emerging technology based system
- o Network based project
- o Any other project based on IT field

• The team will be assigned an internal guide for the project, who will mentor the team during the project execution.

• The team has to take prior permission from the guide for any project that they have chosen.

• The team needs to report to the guide periodically and inform about the project progress. The timings for the reporting time will be informed at the beginning of the semester.

Mini project documentation format:

Acknowledgement

Preface

Index:

1. Project definition
2. Objective and scope of the project
3. Existing system
   a. Introduction
   b. Problem analysis
      i. Study operating problems
      ii. Study informational problems
      iii. Feasibility study
   c. Present system model (Use any design tools / technique)
   d. What's new in the proposed system (if applicable)
4. Proposed system
   a. Functional requirements
   b. Non-functional requirements
   c. Project plan
5. Software requirement analysis and specification
   a. General description
   b. Specific functional and non-functional requirements
6. System design
   a. Proposed system model (Use any design tools / technique)

Development based projects:

7. Development
   a. Source code (for major processes)
8. Implementation
   a. Implementation of the project
   b. Screenshots of the interface
   c. Post-implementation and software maintenance   OR

Analysis based projects:

9. Working outcomes of the existing system
10. Working outcomes of the proposed system
    a. Predicted working outcomes of the proposed system
    b. Comparison of working of proposed system and existing system
11. Technical and managerial lessons learnt
12. Future enhancement
13. References

Semester-VI

Total Lectures: 36   **Total Marks: 50**
                                                             Credits: 2

Learning objective
The programme aims at grooming the participants through sensitizing them about proper behaviour, socially and professionally, in formal and informal circumstances.
The main **objective** of the programme is
- To build self-confidence
- To build enhance self-esteem
- To improve overall **personality** of the participants.

Course outcome
After successful completion of this course student are able to
- Write resume
- Groom corporate habits
- Face frequently asked interview questions
- Understand the importance of respect as a critical corporate value
- Create the right impression in Interviews

Introduction	Introducing the connect with work programme	What is in it for me? Understanding the objective of the CWW programme
Interview Skills	Online Image	Building a strong impression online and sustaining online credibility
	Self-Awareness	To Know your personality through an MBTI
	Grooming	To study corporate grooming habits (The right attire)
	Body Language	To imbibe the right body language for a professional environment
	Confidence	To increase self-belief and faith in one's own abilities
	Interview FAQs	Learn to face frequently asked interview questions
	Resume	To build a strong profiles through effective resume writing
	Rejections	To understand how to handle interview rejections and come back from set-backs
Total no. of Hrs. 13		

Corporate Readiness	Values	An introduction to values in a corporate environment
	Ownership	To learn how to be accountable and own tasks, projects etc.
	Respect	To understand the importance of respect as a critical corporate value
	Teamwork	To understand collaboration and its importance in the corporate world
	Autodidactism	To leverage self-learning and self-directed education
	Flexibility	To learn how to be flexible while playing multiple roles
	Time Management	To improve effectiveness at work and achieve a balance
	Stress Management	To understand how stress can be managed and to lower depression
	Positive Attitude	To take the step towards positive success by adapting the right approach
	LinkedIn (Profile Management)	To gain knowledge on LinkedIn account management and tips to enhance a profile
	SWOT Analysis	To Self evaluate and analyze strengths and areas of improvement
Total no. of Hrs. 15		
Mock Interviews & Group Discussions	In basket simulation (Learning Application)	Group Discussion rules and enhance Public Speaking skills (Group Discussion)
		Tips to handle Interviews and be able to create the right impression (Mock Interviews)
Total no. of Hrs. 8		

<div align="center">

Course Title: Principles of Compiler Design
Course Code: U-PCD-702

</div>

Total Lectures: 50   **Total Marks: 50**
                                                                                                                 Credits: 3

Learning Objectives:
This course will enable a student to:
• Understand the major concept areas of language translation and compiler design
• Develop an awareness of functions and complexity of modern compilers
• Understand various phases of compiler

Course Outcomes
 On successful completion of the course the students will be able to:
• Understand the concrete view on the theoretical and practical aspects of compiler design
• Understand the compilation process
 • Describe the notion of DFA's, NFA's, regular expressions and context free grammar
• Understand and Create lexical rules and grammars for a programming language
• Apply the knowledge primarily obtained by studying structure and context free grammar in the field of designing a compiler for a concise programming language

<div align="center">

SYLLABUS

</div>

UNIT – I   Programming Languages and Compilers
Introduction to Compilers ,Compilers and translators, the structure of compiler, Compiler writing tools, High level programming languages, Definitions of programming languages, A lexical and syntactic structure of a language Data structures, Operators, Statements

UNIT – II   Lexical Analysis
Lexical analysis, Role of a Lexical analyzer, A simple approach to the design of lexical analyzer, Regular expressions, Finite automata, Minimizing number of states of a DFA, Implementation of a lexical analyzer

UNIT – III   Basic Parsing Techniques and Syntax Directed Translation
Context free grammars, Introduction to parsers, Shift reduce parsing, Top-down parsing, Operator Precedence parsing, Predictive parsers, Introduction Syntax Directed Translation, Syntax directed Schemes, Implementation of Syntax directed translators Intermediate code, Postfix notation and evaluation of postfix expressions, Parse trees and syntax trees

UNIT – IV   Symbol Tables, Errors and Code Optimization
 The contents of a symbol table, Data structures for a symbol table, Errors: Lexical-phase errors, Syntactic phase errors, Semantic errors, Introduction Code Optimization, Sources of optimization

Reference books

1. Principles of Compiler Design- By Alfred V. Aho, Jeffrey D. Ullman. Narosa Publishing House ISBN-81-85015-61-9

2. Compilers, Principles, Techniques and Tools - A.V. Aho, Ravi Sethi and J.D. Ullman. ISBN817-808-046-x

3. Introduction to system software By D. M. Dhamdhere

Total Lectures: 50   **Total Marks: 50**
                                                          Credits: 3

Learning Objectives

This course will enable a student to:

- Work With XAMP
- Connect HTML with PHP
- Write Script, Database Connectivity
- Work with Client side and Server side

Course Outcomes

After completion of this course, student will be able to:

- Handle XAMP
- Design form with HTML and write Script
- Write OOP code with php
- Manage State
- Connect Front End with Back End
- Perform DML Operation on Mysql
- Work with Client side and Server side
- Host web pages

SYLLABUS

UNIT – I
Chapter 1   Introduction to PHP
1.1 Evaluation of Php
1.2 Basic Syntax
1.3 Defining variable and constant
1.4 Php Data type
1.5 Operator and Expression

Chapter 2   Handling Html Form with Php
2.1 Capturing Form Data
2.2 Dealing with Multi-value filed
2.3 Generating File uploaded form
2.4 Redirecting a form after submission

UNIT – II

Chapter 3   Function
3.1 What is a function
3.2 Define a function
3.3 Call by value and Call by reference
3.4 Recursive function
3.5 String Related Library function

Chapter 4   Array
4.1 Anatomy of an Array
4.2 Creating index based and Associative array
4.3 Accessing array Element
4.4 Looping with Index based array
4.5 Looping with associative array using each() and foreach()

UNIT – III

Chapter 5   State management
5.1 Using query string(URL rewriting)
5.2 Using Hidden field
5.3 Using cookies
5.4 Using session

Chapter 6   String matching with regular expression
6.1 What is regular expression
6.2 Pattern matching in Php
6.3 Replacing text
6.4 Splitting a string with a Regular Expression

UNIT – IV
Chapter 7   Introduction to OOPS

7.1 Class, Objects, Constructor, Destructor
7.2 Access method and properties using $this variable
7.3 Public, private, protected properties and methods
7.4 Static properties and method
7.5 Inheritance & code reusability
7.6 Polymorphism
7.7 Parent:: & self:: keyword
7.8 Instanceof operator
7.9 Interface

Chapter 8   Database Connectivity with MySql
8.1 Introduction to RDBMS
8.2 Connection with MySql Database
8.3 Performing basic database operation(DML) (Insert, Delete, Update, Select)
8.4 Setting query parameter
8.5 Executing query
8.6 Join
8.7 Web Hosting steps

Reference Books

1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
2. Beginning PHP 5, Wrox publication
3. PHP web services, Wrox publication
4. AJAX Black Book, Kogent solution
5. Mastering PHP, BPB Publication
6. PHP cookbook, O'Reilly publication
7. PHP for Beginners, SPD publication
8. Programming the World Wide Web , Robert W Sebesta(3rd Edition)
9. Check out Joomla presss Pearson (Addison-Wesley Professional).
10. www.php.net.in
11. www.W3schools.com
12. www.wrox.com
13. https://api.drupal.org

<center>**Course Title: Database Administration**
Course Code: U-DBA-704</center>

Total Lectures: 50   **Total Marks: 50**
                                                                       Credits: 3

Learning Objective
To be able to gain suitable expertise in maintenance of a database its availability and to ensure security controls are adequate and are functioning as intended within the operating system.

Course Outcome
On successful completion of the course, the student will:
• Identify and define the information that is needed to design a database management system for a business information problem.
 • Create and maintain databases and tables.
 • Manipulate data in a database using SQL.
 • Manage transactions and locks to ensure data concurrency and recoverability.
• Manage users, privileges and resources
• Start up and shut down an Oracle instance and database
 • Study fundamentals of Recent and Emerging Database Systems in Market.

<center>**Syllabus**</center>

Unit I
1. Basics of DBA
Functions of DBA, Oracle Instance- Starting and Stopping Instance, Memory Architecture- Oracle 10G Memory structure, Background Process, Physical Database Structure -Control file, Data file , Online Redo log file, Archive file, trace file, alert log file, Parameter file SP file password file, Manual Database creation

2. Tablespaces
Introduction to tablespaces - Types of tablespaces - SYSTEM , SYSAUX, Big file, Undo, default, temporary, online, offline tablespaces, read only tablespaces, Working with tablespaces - Creating , Altering and deleting tablespaces, Management of tablespaces

Unit II
3. Physical Database Layouts & storage management.
Traditional disk space storage, Resizing tablespaces and database, Moving datafile, Moving online redo log file, Moving control files, Undo basics - roll back, Read consistency, Database recovery, Flash back operations

4. RAC Database
Overview of Real Application Clusters, RAC database characteristics

Unit III
5. Backup and Recovery
Logical Backup -data pump export/import process, Physical backup - Offline Online backup, Flash Recovery area

6. Database Tuning
Tuning - application design, effective table design, Distribution of CPU requirements, Effective application design, Tuning SQL, Impact of order of load rates, Additional Indexing options, Generating explain plan.

Unit IV
7 Database security & Auditing
Non database security, Database authentication methods, Database authentication, DBA authentication, user and privilege management: Creating database users, Identifying system and object privileges, Granting and revoking privileges, Creating and modifying roles, Auditing

Reference Books
1. Oracle database 10G DBA handbook by Kevin Loney, Bob Bryla Oracle Press
2. OCP oracle database 10G New features for administrator's exam guide By SAM R
Alapati

Total Lectures: 50   **Total Marks: 50**
                                                Credits: 3

Learning Objectives

Students will try to learn:

1. Understand and remember algorithms and its analysis procedure

2. Introduce the concept of data structures through ADT including List, Stack, Queues

3. Design and implement various data structure algorithms and compute the complexity of various algorithms

4. To develop application using data structure algorithms

Course Outcome

Students will able to:

1. Select appropriate data structures as applied to specified problem definition.

2. Implement Linear and Non-Linear data structures operations

3. Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures

4. Implement appropriate sorting/searching technique for given problem and analyze the complexity

UNIT I

Algorithm – Definition and Characteristics, Efficiency of an algorithm, Time and space complexity

Asymptotic notations: big-oh notation (O), big-omega notation (Ω), big-theta notation (θ) , Time space trade-off , Abstract data type,

Introduction to data structures: Need of data structure, Categories of data structures, Operations, Arrays storage representation of 1D, 2D and Multi-dimensional arrays, Sparse matrix, operations and Representation, Applications of array

UNIT II

Stacks & Queues: Stack - Definition, Operation on stack, Implementation using arrays and linked lists, Applications of Stacks, Function Calling, Recursion- direct & indirect recursion, Evaluation of arithmetic Expressions, Conversion of Expressions - Prefix, Infix and Postfix expressions. Queues: Definition, Implementations using arrays and linked lists, Circular queue, Dequeue, Priority queues, Applications of queues.

UNIT III

Linked list: Concept of linked list, Operations on Linked list: Creation, insertion, Deletion, Insertion and deletion examples, Types of linked list: Circular linked list, Doubly linked list, Implementation of stack and queue using linked list

Trees Graphs and their applications

Trees terminology, Binary tree, Traversal methods, Tree traversal algorithms, Threaded binary tree, Graph terminologies, Graph Representation: Sequential and Linked list, Graph Traversal – Depth First Search, Breadth First Search, Applications, AOV network, topological sort

UNIT IV

 Searching: Linear and Binary

Sorting: Bubble sort, Selection sort, Insertion sort

More algorithms: Divide and Conquer, Greedy Method, Dynamic Programming, Backtracking

Reference books

1. Data Structure using C by A.M. Tanenbaum, Yecidyan lang

2. Data Structure Through C- By Dr. Sahani**.**

3. Data Structures Using C Yashwant Kanitkar – BPB Publication

4. Data Structures Through C (A Practical Approach), G.S Baluja Danapat Rai & Co.

5. Fundamentals of Data Structures, Ellis Horowitz and Sartaj Sajni Galgotia Publications

6. Introduction to data structures in C , Ashok N. Kamthane, Pearson Education

7. Theory and Problems of Data Structures, Schaums Outline Series, Seymour Lipschutz

<div align="center">
Course Title: Data Mining
Course Code: U-DAM-706
</div>

Total Lectures: 50   **Total Marks: 50**
                                                               Credits: 3

Learning Objectives
 This course will enable a student to:
1. Understand the concept of data mining as an important tool for enterprise data management and as a cutting edge technology for building competitive advantage.
2. Effectively identify sources of data and process it for data mining
3. Design and implement all data mining algorithms and also impart knowledge of tools used for data mining
5. Provide knowledge on how to gather and analyze large sets of data to gain useful business understanding.
6. Impart skills for analyzing business problems by identifying opportunities

Course Outcome
After completion of this course, students will able to:
 1. Understand the importance of data mining and the principles of business intelligence
2. Organize and prepare the data needed for data mining using pre preprocessing techniques
 3. Perform exploratory analysis of the data to be used for mining.
4. Implement the appropriate data mining methods like classification, clustering or Frequent Pattern mining on large data sets.
 5. Define and apply metrics to measure the performance of various data mining algorithms.

<div align="center">

Syllabus
</div>

 Unit -I
Chap-1 Introduction
What is Data Mining?, Definition, DBMS Vs Data Mining, Issues and Challenges in DM, DM Application Areas, , Basic Data Mining Task, Data mining metrics, Social implications of data mining, Data Mining Vs Knowledge Discovery in Databases(KDD)

Chap-2 Related concepts
OLTP system, IR system, Decision Support Systems, Dimensional modeling, Data Warehousing, OLAP, Web Search Engines

Unit –II
Chap -3 Statistical perspectives on data mining

Point estimation, Model based on summarization, Bayes Theorem, Hypothesis testing, Regression and correlation

Chap-4 Non parametric techniques
Decision trees, Neural networks, Genetic algorithms

Unit-III
Chap 5 Classification
Introduction, Issues in classification, Bayesian classification, Distance based algorithms: KNN, Decision tree, NN supervised learning

Chap 6 Clustering
Introduction, Outliers, Hierarchical algorithms, Partitioned algorithms, BRICH algorithm

Unit IV
Chap 7 Association
What is an Association rule?, Method to discover Association Rule, Basic algorithms, Large item sets, Data parallelism

Reference books
1. Data Mining Techniques : Arun K. Pujari ,
2. Data Mining: Introductory and Advanced Topics: M. H. Dunham Pearson Education.
3. Data Mining: Concepts & Techniques, Morgan Kaufman. 2006

Text Books
1. Jiawei Han, Micheline Kamber, "Data mining: concepts and techniques", Morgan Kaufmann Publisher, second edition
2. G. K. Gupta, "Introduction to Data mining with case studies", PHI, second edition

<div align="center">

Course Title: Cloud Computing
Course code: U-CLC-707

</div>

Total Lectures: 50   **Total Marks: 50**
                                                                Credits: 3

Learning Objectives
This course will enable students to acquire the knowledge of cloud computing fundamentals, architecture, services, implementation and deployment techniques, security challenges and future research trends.

Course Outcome
1. After completion of the course the student should be able to:
2. Differentiate various computing techniques
3. Analyze the performance, scalability, and availability of the underlying cloud technologies and software
4. Compare various cloud computing providers/software's
5. Implement and administrate Cloud and Mobile Cloud Security.
6. Compare the advantages and disadvantages of various cloud computing platforms.
7. Identify security and privacy issues in cloud computing.
8. Solve a real-world problem using cloud computing through group collaboration.

Syllabus

UNIT I
1. Introduction of Cloud Computing
    1.1.   Introduction
    1.2.   Cloud computing compared with Virtualization
    1.3.   Benefits of cloud computing
    1.4.   Client server model

2. Cloud Deployment Models
    2.1 Private cloud
    2.2 Public Cloud
    2.3 Hybrid Cloud
    2.4 Community cloud
    2.5 Popularity of Cloud deployment models

UNIT – II
3. Cloud Service Models
    3.1 Various Cloud Services
    3.2 PaaS
        I.   Model
        II.   Architecture
        III.   Platform as a service: Google App Engine

3.3 SaaS
    I.   Model
    II.   Architecture
    III.   Software as a service: Microsoft Azure
3.4 IaaS
    I.   Model
    II.   Architecture
    III.   Infrastructure as a service: Amazon EC2

UNIT III
4. Cloud Deployment Techniques
    4.1 Factors for Successful Cloud Deployment
    4.2   Network Requirements
    4.3   Potential Problem areas in a cloud network and their Mitigation
    4.5 Cloud Network Topologies

5. Cloud Technologies: Web Services, AJAX and MASHUPs
    5.1 web services: SOAP and REST
    5.2 AJAX: asynchronous 'rich' interfaces
    5.3 Mashups: User interface services

UNIT IV
6. Mobile Cloud Computing
    6.1 Introduction
    6.2 Benefits of MCC
    6.3 Architecture of MCC
    6.4 challenges in mobile and at cloud shield

7. Cloud and Mobile Cloud Security Challenges
    7.1 Benefits for Security as a service
    7.2   Issues with Security as a Service
    7.3 Cloud Security Architecture
    7.4 Mobile Cloud Security Architecture

Reference Books
    1. Enterprise Cloud Computing: Technology, Architecture, Application By Gautam Shroff
    2. Cloud Computing Principles and Paradigms, Rajkumar Buyya Wiley
    3. Distributed and Cloud Computing, Kai Hwang, Mk Publication
    4. Cloud computing Black Book Dreamtech Publication

Course Title: Lab Course XXIII (Compiler design)
Course Code: U-LAC-708

Total Marks: 50   **Credits: 2**

Learning Objectives
 This course will enable a student to:
• Understand the major concept areas of language translation and compiler design
• Develop an awareness of the function and complexity of modern compilers.
• Understand various compilation phases and their working

Course Outcomes
After completing this course, student will be able to:
 • Understand the concrete view on the theoretical and practical aspects of compiler design
• Understand the compilation process
 • Describe the notion of DFA's, NFA's, regular expressions and context free grammar
• Understand and Create lexical rules and grammars for a programming language
• Apply the knowledge primarily obtained by studying structure and context free grammar in the field of designing a compiler for a concise programming language

Proposed Practical List

1. Program to demonstrate operations on stack and queue data structure

2. Program to demonstrate lexical analysis phase

3. Program to demonstrate working of syntax analyzer

4. Program to demonstrate semantic analysis phase

5. Program to demonstrate intermediate code generation phase

6. Program to demonstrate code optimization phase

7. Program to demonstrate code generation phase

8. Program to demonstrate symbol table

9. Program to demonstrate error handler phase

10. Program to design and simulate finite automata

Course Title: Lab Course XXIV (Web Programming Using PHP)
Course Code: U-LAC-709

Total Marks: 50   **Credits: 2**

Learning Objectives
This course will enable a student to:
- Work With XAMP
- Connect HTML with PHP
- Write Script, Database Connectivity
- Work with Client side and Server side

Course Outcomes
After completion of this course, student will be able to:
- Handle XAMP
- Design form with HTML and write Script
- Write OOP code with php
- Manage State
- Connect Front End with Back End
- Perform DML Operation on Mysql
- Work with Client side and Server side
- Host web pages

Proposed Practical List

1. Hello World program
2. Script to Capture Form Data
3. Script to Redirect form
4. Script to define and call function
5. Script to work with String function
6. Script for Index Based Array
7. Script for Associative Array
8. Script for foreach loop
9. Script to work with Query String
10. Script to work with Hidden Field
11. Script to work with Cookie
12. Script to work with Session
13. Script to demonstrate Regular Expression
14. Script to demonstrate Inheritance

15. Script to demonstrate Interface
16. Script to demonstrate Constructor and Destructor
17. Script to connect with Mysql database
18. Script to perform DML operation on database
19. Script to perform show database record on web page
20. Study of Web Hosting

Total Marks: 50	Credits: 2

Learning Objective
To be able to gain suitable expertise in maintenance of a database its availability and to ensure security controls are adequate and are functioning as intended within the operating system.

Course Outcome
On successful completion of the course, the student will:
• Identify and define the information that is needed to design a database management system for a business information problem.
 • Create and maintain databases and tables.
 • Manipulate data in a database using SQL.
 • Manage transactions and locks to ensure data concurrency and recoverability.
• Manage users, privileges and resources
• Start up and shut down an Oracle instance and database
 • Study fundamentals of Recent and Emerging Database Systems in Market.

Proposed Practical List:

1. Review of SQL commands using graphical environment

2. Study of role of DBA

3. Creating a Database

4. Starting and stopping database instance

5. Study of tablespace management

6. Administering the Control Files and Redo Logs

7. Study of Tables, Indexes and Constraints

8. Study of Backup and recovery using RMAN

9. Study of Rollback Segments

10. Study of user management: creation, alteration and deletion

11. Granting and revoking privileges to users

12. Study of role management

Course Title: Lab Course XXVI (Data Structure)
Course Code: U-LAC-711

Total Marks: 50   **Credits: 2**

Learning Objectives

This course enables the student to be able to:

1. Understand, remember and compute the complexity of various algorithms

2. Design and implement various data structure algorithms

3. Introduce various techniques for representation of the data in the real world

4. Develop application using data structure algorithms

Course Outcome

After completion of this course, students will able to:

1. Select appropriate data structures as applied to specified problem definition.

2. Perform operations like searching, insertion, and deletion, traversing mechanism on various data structures.

   3. Implement appropriate sorting/searching technique for given problem.

4. Determine and analyze the complexity

Proposed Practical list

1) Implement Linear Search algorithm to find an item in a list.

2) Implement Binary search algorithm to find an item in an ordered list.

3) Implement Sorting Algorithms

       a. Bubble sort

       b. Insertion sort

       c. Selection sort

4) Implement insertion and deletion algorithms of Stack

5) Implement Program for:

       a. Infix to Postfix conversion

       b. Postfix Evaluation

6) Implement the following

       a. Simple queue

       b. Circular queue

7) Implement Linked list algorithm for insertion and deletion of an item from list

8) Implement Binary Tree traversal algorithms

9) Recursive implementation of

      a. Factorial

      b. Fibonacci

      c. Tower of Hanoi

Course Title: Lab Course XXVII (Data Mining)
Course Code: U-LAC-712

Total marks: 50   Credits: 2

Learning Objectives
 This course will enable a student to:
1. Understand the concept of data mining as an important tool for enterprise data management and as a cutting edge technology for building competitive advantage.
2. Effectively identify sources of data and process it for data mining
3. Design and implement all data mining algorithms and also impart knowledge of tools used for data mining
5. Provide knowledge on how to gather and analyze large sets of data to gain useful business understanding.
6. Impart skills for analyzing business problems by identifying opportunities

Course Outcome
After completion of this course, students will able to:
 1. Understand the importance of data mining and the principles of business intelligence
2. Organize and prepare the data needed for data mining using pre preprocessing techniques
 3. Perform exploratory analysis of the data to be used for mining.
4. Implement the appropriate data mining methods like classification, clustering or Frequent Pattern mining on large data sets.
 5. Define and apply metrics to measure the performance of various data mining algorithms.

Proposed practical list:

Practicals will be based on Unit I/II/III/IV with the help of WEKA tool for demonstrating various data mining concepts.

Course Title: Lab Course XXVIII (Cloud Computing)
Course Code: U-LAC-713

Total Marks: 50   **Credits: 2**

Learning Objectives

This course will enable students to acquire the knowledge of cloud computing fundamentals, architecture, services, implementation and deployment techniques, security challenges and future research trends.

Course Outcome

After completion of the course the student should be able to:

1. Differentiate various computing techniques
2. Analyze the performance, scalability, and availability of the underlying cloud technologies and software
Compare various cloud computing providers/software's
3. Implement and administrate Cloud and Mobile Cloud Security.
4. Compare the advantages and disadvantages of various cloud computing platforms.
5. Identify security and privacy issues in cloud computing.
6. Solve a real-world problem using cloud computing through group collaboration

Proposed Practical's List

1. Study of Cloud Computing & Architecture
2. Study of Virtualization in Cloud
3. Study and implementation of Infrastructure as a Service
4. Study and installation of Storage as a Service
5. Study and implementation of Platform as a Service
6. Study of Securing Servers in Cloud, data directory and introduction to encryption for own cloud.
7. Study of administrative features: Cloud Management ,User Management
8. Case study on Amazon EC2
9. Case study on Microsoft azure
10. Case study on Google App. Engine
11. Mini Task: Case Study on any open system used for cloud.

<div align="center">

Course Title: Major Project
Course Code: U-PRW-641

</div>

Total Marks: 100   **Credits: 4**

Learning objectives
- Discover potential research areas in the field of IT
- Identify real world problems and challenges that need IT based solutions
- Demonstrate an ability to work in teams
- Improve the team building, communication and management skills of the students.
- Conduct a survey of several available literature in the preferred field of study
- Use all concepts of IT in creating a solution for a problem
- Formulate and propose a plan for creating a solution for the research plan identified
- Report and present the findings of the study conducted in the preferred domain

Course outcome
After completing this course, the students will be able to have hands-on to carry out the Project Work using various platforms and project tools which will provide practical experience
- Implement what is learnt during course
- Get real-life experience by working in real systems
- Understand the problems faced during project implementation.
- Enhance the problem solving ability by solving the real-time problems.
- Learn team work and appreciate role of each of the team members.

Major project Description:

• A team of maximum 3 students can be formed for this project and work together to learn working collaboratively.

• The team can choose the project area from the following:
  - Java based project
  - .Net based project
  - A project consisting analysis of an existing emerging technology based system
  - Network based project
  - Any other project based on IT field

• The team will be assigned an internal guide for the project, who will mentor the team during the project execution.

• The team has to take prior permission from the guide for any project that they have chosen.

• The team needs to report to the guide periodically and inform about the project progress. The timings for the reporting time will be informed at the beginning of the semester.

Major project documentation format:

Acknowledgement

 Preface

Index:

1. Project definition

2. Objective and scope of the project

3. Existing system

       a. Introduction

       b. Problem analysis

            i. Study operating problems

            ii. Study informational problems

            iii. Feasibility study

       c. Present system model (Use any design tools / technique)

       d. What's new in the proposed system (if applicable)

4. Proposed system

       a. Functional requirements

       b. Non-functional requirements

       c. Project plan

5. Software requirement analysis and specification

       a. General description

       b. Specific functional and non-functional requirements

6. System design

       a. Proposed system model (Use any design tools / technique)

Development based projects:

7. Development

       a. Source code (for major processes)

 8. Implementation

       a. Implementation of the project

        b. Screenshots of the interface

        c. Post-implementation and software maintenance   OR

Analysis based projects:

 9. Working outcomes of the existing system

 10. Working outcomes of the proposed system

       a. Predicted working outcomes of the proposed system

       b. Comparison of working of proposed system and existing system

11. Technical and managerial lessons learnt

12. Future enhancement

13. References