

**PUNYASHLOK AHILYADEVJI HOLKAR
SOLAPUR UNIVERSITY, SOLAPUR**



Name of the Faculty: Science & Technology
CHOICE BASED CREDIT SYSTEM

Syllabus: Computer Science

Name of the Course: B.Sc. II (Sem-III & IV)
(w.e.f. June 2020)

PUNYASHLOK AHILYADEVI HOLKAR SOLAPUR UNIVERSITY, SOLAPUR

Syllabus for B.Sc. Part – II (CBCS Semester Pattern)

Computer Science

(W.E.F. June 2020)

Introduction

B.Sc-II (Computer Science) CBCS Semester wise pattern to be introduced from June 2020.

The Details of syllabus is as follows.

Semester No.	Type of the Paper	Paper No.	Title of the Paper	University Exam.	Internal Exam.	Total	Credits
III	DSC 1C	Paper-V	Data Structure	40	10	50	4
		Paper-VI	Design analysis and algorithm	40	10	50	
	SEC -I		AngularJS	40	10	50	2
IV	DSC 1D	Paper-VII	Software Engineering	40	10	50	4
		Paper-VIII	Database Management System	40	10	50	
	SEC -II		PHP	40	10	50	2
Annual Examination		Practical	Paper -II (Practical on paper V)	40	10	50	4
			Paper -III (Practical on paper VIII)	40	10	50	
Total (Without SEC-I & SEC-II)				240	60	300	12

Note: Nature of internal examination, passing standard, ATKT and the conversion of marks into grades and credits are as per guidelines of Science Faculty Credit and Grading System.

Teaching Periods:

1. Total teaching periods for each theory paper (Paper-V to VIII) is three periods per week
2. Total teaching periods for each practical Paper- II and Paper-III are four periods per week per paper per batch.
3. Total Teaching periods for SEC is 2.5 periods per week

Equivalence papers for B.Sc-II Sem III and IV (Computer Science)

Sr.no.	Old Paper	New Paper
1	Object oriented programming using C++	No equivalence
2	Software Engineering	Paper VII-Software Engineering
3	Data structure	Paper V-Data Structure
4	DBMS using oracle	Paper VIII-Database Management System

Semester - III

Computer Science Paper - V: Data Structures

Course objectives:

1. To impart the basic concepts of data structures and algorithms
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs
4. To impart the basic concepts of data structures and algorithms

Unit 1 **(15)**

Introduction of Data Structure, Need of Data Structure, Types of Data Structure, ADT, Stack: Introduction to stack, Representation-static & dynamic, stack Operations, Application -infix to postfix & prefix, postfix evaluation, recursion, expression validity. Queues: Introduction to Queue, Representation -static & dynamic, Operations, Circular queue, Double ended queue, priority queues, Applications of Queue.

Unit 2 **(15)**

Linked List:-Introduction to List, Implementation of List - static & dynamic representation, Types of Linked List, Operations on List, Applications of Linked List - polynomial manipulation

Trees: Concept & Terminologies, Binary tree, binary search tree, Representation - static & dynamic, Operations on BST - create, Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes, Height balance tree- AVL, B tree, B+ Tree,

Graph- Graph terminology, Representation of graphs, Graph Traversal-BFS (breadth first search), DFS (depth first search), Minimum spanning Tree

Unit 3

(15)

Sorting: Bubble sort, Quick sort, Simple Insertion sort, Shell sort, Address calculation sort, Selection Sort, Heap Sort, Merge sort, Radix Sort.

Searching: Linear Search, Binary Search, and Tree searching methods, Multiway search tree, Hash function (open and close).

Course Outcome:-

After learning the course the students should be able:

1. Differentiate primitive and non-primitive structures
2. Choose appropriate data structures and algorithms, understand the ADT/libraries, and use it to design algorithms for a specific problem
3. Design and apply appropriate data structures for solving computing problems.
4. Apply sorting and searching algorithms to the small and large data sets.

Reference Books

1. Data Structures and Algorithms, Pearson Education, Reprint 2006 by Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman
2. Algorithms, data structures, Programs by Niklaus Wirth:
3. File Systems, Structures and Algorithms (PHI). By Thoms Horbron:
4. Art of computer Programming Vol - I. by D. E. Kunth:
5. Data structures using C and C++ (PHI). By Tanenbaum:
6. Fundamentals of computer algorithms by 2nd edition galgotia publication by Ellis horowitz, sartaj sahni

Computer Science Paper – V: Design and Analysis of Algorithms

Course objectives:

1. To understand and apply the algorithm analysis techniques.
2. To critically analyze the efficiency of alternative algorithmic solutions for the same problem
3. To understand different algorithm design techniques.
4. To understand the limitations of Algorithmic power.

Unit 1

(15)

What is an algorithm?, Algorithm: Definition, characteristics, Space complexity, time complexity, Asymptotic notation (Big O, Omega Ω , theta Θ , Fundamentals of Algorithmic Problem Solving , Important Problem Types, Fundamentals of the Analysis of Algorithmic Efficiency , The efficient algorithm, Average, Best and worst case analysis, Amortized analysis , Asymptotic Notations properties, Analyzing control statement, Loop invariant and the correctness of the algorithm. Analysis Framework, Empirical analysis, Mathematical analysis for Recursive and Non-recursive algorithms, Visualization

Unit 2

(15)

Brute Force – Computing an – String Matching – Closest-Pair and Convex-Hull Problems – Exhaustive Search – Travelling Salesman Problem – Knapsack Problem – Assignment problem. Divide and Conquer Methodology – Introduction, Recurrence and different methods to solve recurrence, Multiplying large Integers Problem, Problem Solving using divide and conquer algorithm - Binary Search, Max-Min problem, Sorting (Merge Sort, Quick Sort), Matrix Multiplication, Exponential, Multiplication of Large Integers – Closest-Pair and Convex – Hull Problems.

Dynamic programming – Principle of optimality – Coin changing problem, Computing a Binomial Coefficient – Floyd’s algorithm – Multi stage graph – Optimal Binary Search Trees – Knapsack Problem and Memory functions. Greedy Technique – Container loading problem – Prim’s algorithm and Kruskal’s Algorithm – 0/1 Knapsack problem, Optimal Merge pattern – Huffman Trees, All Points Shortest path, Matrix chain multiplication, Longest Common Subsequence.

Unit 3

(15)

The Simplex Method – The Maximum-Flow Problem – Maximum Matching in Bipartite Graphs, Stable marriage Problem

Backtracking and Branch and Bound Introduction, The Eight queens problem , Minimax principle, String Matching:-Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm.

Lower – Bound Arguments – P, NP NP- Complete and NP Hard Problems. Backtracking – n-Queen problem – Hamiltonian Circuit Problem – Subset Sum Problem. Branch and Bound – LIFO Search and FIFO search – Assignment problem – Knapsack Problem – Travelling Salesman Problem – Approximation Algorithms for NP-Hard Problems – Travelling Salesman problem – Knapsack problem.

Course Outcome:-

At the end of the course, the student should be able to:

1. Design algorithms for various computing problems.
2. Analyze the time and space complexity of algorithms.
3. Critically analyze the different algorithm design techniques for a given problem.
4. Modify existing algorithms to improve efficiency.

Reference Books

1. Introduction to the Design and Analysis of Algorithms, Third Edition, Pearson Education, 2012. By Anany Levitin
2. Introduction to Algorithms, Third Edition, PHI Learning Private Limited, 2012 by Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein
3. Data Structures and Algorithms, Pearson Education, Reprint 2006 by Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman
4. The Algorithm Design Manual, Second Edition, Springer, 2008. By Steven S. Skiena,
5. Fundamental of Algorithms by Gills Brassard, Paul Bratley, PHI.
6. Foundations of Algorithms, by Shailesh R Sathe, Penram
7. Design and Analysis of Algorithms, Dave and Dave, Pearson.

Computer Science Paper - SEC -I- AngularJS

Course objectives:

1. Understanding Basic concept of AngularJS
2. Properly separate the model, view, and controller layers of your application and implement them using AngularJS
3. Master AngularJS expressions, filters, AngularJS directives and scopes
4. Build Angular forms
5. Understand the design of single-page applications and how AngularJS facilitates their development

Unit-1

(10)

What is AngularJS?, Why AngularJS?, Features of AngularJS, AngularJS architecture, Setting up the Environment, Model-View-Controller explained, My first AngularJS app
All about Angular expressions, How to use expressions, Number and String Expressions, Object Binding and Expressions, Working with Arrays, Forgiving Behaviour, Angular expressions v/s Javascript expressions

Unit-2

(20)

Built-in filters, Uppercase and Lowercase Filters, Currency and Number Formatting Filters, OrderBy Filter, Filter Filter, Using AngularJS filters, Creating custom filters
Introduction to AngularJS Modules, Module Loading and Dependencies, Creation vs Retrieval, Bootstrapping AngularJS
Role of a Controller, Attaching properties and functions to scope, Nested Controllers, Using filters in Controllers, Controllers in External Files, Controllers & Modules, Controllers
Introduction to Directives, Directive lifecycle, Using AngularJS built-in directives, Core Directives, Conditional Directives, Style Directives, Mouse and Keyboard Events Directives, Matching directives, Creating a custom directive

Unit-3

(15)

Working with Angular Forms, Model binding, Understanding Data Binding, Binding controls to data, Form controller, Validating Angular Forms, Form events, Updating models with a twist, \$error object

What is scope, Scope lifecycle, Two way data binding, Scope inheritance, Scope & controllers, Scope & directives, \$apply and \$watch, Rootscope, Scope broadcasting, Scope events

Single Page Application (SPA)-what is SPA, Pros & Cons of SPA, Installing the ngRoute module, Configure routes, Passing parameters, Changing location, Resolving promises, Create a Single Page Application

AngularJS Animation - ngAnimate Module, CSS transforms, CSS transitions, Applying animations, Directives supporting animation

Course Outcome:-

At the end of the course, the student should be able to:

1. Build an awesome User Interface
2. Create and bind controllers with JavaScript
3. Validate user input data
4. Write own filters, directives and controls
5. Create animation in web page and Create single page application

Reference Books:

1. Professional AngularJS by Diego Netto and Valeri Karpov-Wrox press
2. Learning AngularJS by Brad Dayley- Addison-Wesley Professiona
3. AngularJS by Brad Green and Shyam Seshadri- O'Reilly
4. Pro AngularJS by Adam Freeman-APress
5. Learning AngularJS: A Guide to AngularJS Development by Ken Williamson-O'Reilly Media
6. Beginning AngularJS by Andrew Grant-Apress

Semester - IV

Computer Science Paper - VII: Software Engineering

Course objectives:

1. To study fundamental concepts in software engineering, SDLC, software requirements specification, formal requirements specification and verification.
2. To study the basic techniques for improving quality of software.
3. Understand the fundamental principles of Software Engineering & will also have a good knowledge of responsibilities of project manager and how to handle these.
4. To understand the basic knowledge of different models.

Unit 1: (15)

System concepts: Introduction, Definition, Elements of system, system concepts, Types of system, System Analysis, Role of System Analyst.

Software Engineering: Definition, Characteristics of software, Qualities of software. System Development life cycle, Process Models-Waterfall model, v shape model, Spiral model, Prototyping, incremental, RAD

Unit 2: (15)

Requirement Analysis, Fact finding techniques: Interviews, Questionnaire, Record reviews, Observation, Basic and User design requirements, Organization Dependent Requirements,

Analysis and Design Tools: Flow charting, Decision tables, Decision Trees, Structure charting Techniques (HIPO).

System Design: Data flow Diagram (Physical, Logical), Entity relation diagram, Input output design, structured chart

Unit 3:

(15)

Configuration of the System: Collection of system statistics, Setting Sub-system Boundaries. Construction of the system: traditional and incremental approaches, conversion methods, Software Testing: Need of Testing, types of testing, Software Implementation and maintenance, System Development Tools: Role, Benefits and weakness of case Tools, Taxonomy of case tools, Case studies: Pay Roll, Fixed Deposit, Inventory system, College Admission System, Library System, Loan system.

Course Outcome:-

At the end of the course, the student should be able to:

1. Basic knowledge and understanding of the analysis and design of software systems.
2. Ability to apply software engineering principles and techniques to develop, maintain and evaluate large-scale software systems.
3. To produce efficient, reliable, robust and cost-effective software solutions.
4. Ability to perform independent research and analysis.
5. Ability to work as an effective member or leader of software engineering teams.

Reference Books:

1. Analysis and Design of Information Systems By James Senn.
2. System analysis and Business application (for case studies) By Rajesh Nike / swapna kishore.
3. Software Engineering By Pressman.
4. System Analysis and Design By Parthsarty / Khalkar.
5. Practical guide to structure System Design By Miller/Page/jones.

Computer Science Paper – VIII: Database Management System

Course objectives:

1. To understand the fundamental concepts of database.
2. It helps in developing skills for the design and implementation of a database applications
3. To understand user requirements and frame it in data model.
4. To understand creations, manipulation and querying of data in databases.
5. Undersetting SQL and PL/SQL

Unit 1:

(15)

Introduction to database system:-Definition, Limitations of traditional file system, Advantages of DBMS, Components of DBMS, Database Architecture, Database Users, Schemas and instances, 2 tier and 3 tier architecture, Database languages, Types of data models- relational, Network, Hierarchical, Distributed

E-R model: E-R Diagram, entities, attributes and its types, Relationship and relationship sets, Cardinality, Degree, Generalization, Specialization, Aggregation

Relational Model and Database design:-Relation, Domain, Tuples, types of keys, relational integrity rules, Dr. Codd's rules,

Relational Algebra operations:- Select, Project, Cartesian Product, Union, Set difference, Natural Join, Outer Join,

Dependencies and its types, Normalization and its types-1NF, 2NF, 3NF, BCNF, lossless joins, Data dictionary. Case study

Unit 2:

(10)

Transaction Management & Concurrency Control: -Introduction, Definition, properties, transaction states, scheduling and its types, conflict and view serializability,

Introduction to Concurrency Control, problems of concurrency control. lock based protocols, timestamp based protocol, deadlock, deadlock handling.

Database recovery and Atomicity:-Introduction, recovery algorithms, log base recovery, shadow paging, checkpoints or syncpoints or savepoints.

Unit 3:

(20)

History of MySQL, Installation of MYSQL, MySQL Architecture, Invoking MySQL through Command Line, MySQL Server Start and Stop, Overview of Data Types in MySQL, Defining a Database, Creating Tables and Fields in MySQL

SQL & Procedural Language in MySQL:-DDL, DML,DCL queries, Simple Queries, Expressions, Conditions and Operators, Functions, Groupby-having, Where clause, Joins, Sub queries, Views, indexes, sequences. Blocks, Conditional statement and loops, Cursors and types, procedures and functions, packages, trigger, Exception Handling.

Course Outcome:-

At the end of the course, the student should be able to:

1. Demonstrate the concepts of Relational database model , ER model and Distributed databases.
2. Design E-R Model for given requirements and convert the same into database tables
3. Implement database operations and transactions using SQL.
4. Apply the concepts of Transaction processing, Concurrency control, Database Recovery and Back-up in applications.
5. Use database techniques such as SQL & Procedural Language in MySQL.

Reference Books:

1. Database System Concepts By KorthSilberschetz
2. Fundamentals of Database Systems by Elmsari, Navathe
3. Teach Yourself SQL in 14 Days by Jeff Parkins and Bryan Morgan
4. An Introduction to Database Systems by Bipin Desai
5. SQL and PL/SQL Programming by Ivan Bayross
6. SQL and PL/SQL Programming by Oracle Press
7. MySQL(TM): The Complete Reference by Vikram Vaswani
8. MySQL Cookbook by Paul DuBois

Computer Science Paper – SEC –II-PHP

Course objectives:

1. To understand the fundamental concepts of PHP.
2. To acquire knowledge and skills for creation of web site considering both client and server side To gain ability to develop responsive web applications
3. To understand OOP concepts with PHP.
4. Able to develop Dynamic web applications

Unit-1

(15)

Introduction, Basics of PHP: History of PHP, features of PHP, Interfaces to External systems, Hardware and Software requirements, Benefits of PHP as a server side languages, How PHP works with the web server, Installation and Configuration files, PHP Framework, Basic PHP syntax, PHP data types, Displaying type information, Testing for specific data type, Changing type with Set type, Operators, Variable manipulation, Dynamic variables, Static vs. Dynamic Optimization, Redirecting web pages

Control Structures If condition Statement, The switch statement, Using the ? operator, While, do while and for Loop, Breaking out of loops, Nesting loops

Array, String and Functions:, Array: Single-Dimensional Arrays , Multidimensional Arrays, Associative arrays, Accessing arrays, Getting the size of an array, Examining arrays, merging arrays, Sorting arrays, Sorting an associative arrays, String:, Formatting String for Presentation, Formatting String for Storage Joining and Splitting String, Comparing String, Matching and replace Substring, patterns, The basic regular expressions, Matching patterns, Finding matches

Function :- Function and its Types , Library Function, Array functions, String functions, Date and time functions, Maths functions, User-defined functions, Creating a function, Returning value from function, Dynamic function calls, Variable scope, Accessing variable with the global statement, Function calls with the static statement, Setting default

values for arguments, Passing arguments to a function by value, Passing arguments to a function by reference, Using require() and include()

Unit-2 **(20)**

Object Oriented Programming in PHP: Object oriented concepts, Define a class and objects, Class attributes, Object properties, Object methods, constructors and destructors, Class constants, Static method, inheritance, Abstract classes, Exception Handling, Final keyword, Implementing Interface, Object serialization, Understanding Advance and New, Checking for class and method existence, File Handling-reading and writing file.

Unit-3 **(10)**

Working With Forms and Database (MySQL):Working With Forms:, Forms, Forms controls properties, methods and events, Retrieving form data with \$_POST, \$_GET and \$_REQUEST arrays, Validating retrieved data, Strategies for handling invalid input, Super global variables, Super global array, Importing user input, Accessing user input, Combine HTML and PHP code, Using hidden fields, Redirecting the user, File upload and scripts, Validation :-Server side validation, Client side validation (Java script)

Working with Database MySQL:-Working with PHP-MySQL Environment, Connecting to the MYSQL, Selecting a database, Adding data to a table, Displaying returned data on Web pages, Finding the number of rows, Inserting, deleting and updating data, Executing multiple queries,

State Management- Cookies: What is a Cookie?, Setting time in a cookie with PHP, Deleting a cookie, Creating session cookie, Working with the query string, Session:- What is session?, Starting a session, Registering Session variables, working with session variables , destroying session, passing session Ids, encoding and decoding session variables.

Course Outcome:-

At the end of the course, the student should be able to:

1. Write PHP scripts to handle HTML forms.
2. Create PHP programs that use various PHP library functions, and that manipulate files and directories.
3. Validate user input
4. Analyze and solve various database tasks using the PHP language.
5. Analyze and solve common Web application tasks by writing PHP programs.

Reference Books:

1. PHP: The Complete Reference-Steven Holzner.
2. Professional PHP 5-Ed Lecky-Thompson, HeowEide-Goodman, Steven D. Nowicki, Alec Cove.
3. Programming PHP-Rasmuslerdorf, Kevin Tatroe.
4. Beginning PHP 5.3 -Wrox Plublication-Matt Doyle
5. Learning php, mysql, javascript and css -Oreilly-Robin Nixon

Sample Practical Assignment on Data Structure

1. Write a program on recursive function
2. Write a program to implement stack using array.
3. Write a program to implement stack using list.
4. Write a program to check whether the expression is valid or not.
5. Write a program to convert infix expression to postfix.
6. Write a program to implement queue using static.
7. Write a program to implement queue using dynamic method.
8. Write a program to implement Circular queue.
9. Write a menu driven program to implement singly, doubly, singly circular , doubly circular linked list with operation a) insert at beginning of linked list b) insert at specific position c) insert at end d) delete the first node e) delete specific node f) delete last node g) display the list
10. Write a menu driven program to implement singly linked list with operation a) sort list b) maximum value c) minimum value d) find & replace a value e) count the number of nodes.
11. Write a program to create binary search tree and display its contents by using inorder, preorder and postorder traversal method.
12. Write programs to implement a) Bubble Sort Technique. b) Straight Selection Sort Technique. c) Simple Insertion Sort Technique. d) Shell Sort Technique. e) Quick Sort Technique. f) heap sort technique. g) address calculation Sort Technique. h) Radix Sort Technique. i) Merge Sort Technique.
13. Write programs to implement a) sequential searching Technique. b) indexed searching Technique. c) Binary searching technique
14. Write a program to add, subtract two polynomials by using Linked list
15. Write a program to implement Graph traversing technique.

Sample Practical Assignment on Database Management System

1. Create table employee (eno, name, dept, basic salary, HRA, tax, deduction). Dept are D1, D2, D3 and D4. Use constraints.
 - a. Insert 20 records.
 - b. Display total amount spend by company on salary.
 - c. Display name of dept for which company spend maximum amount.
 - d. Display average salary of employee in company.
 - e. Display average salary of each dept.
 - f. Display total salary for each dept.
 - g. Display highest salary for each dept.
 - h. Display different between average of max salary for each dept and average of each dept.
 - i. Display no of dept in the company.
 - j. Display name of all employee whose basic pay is higher then average salary.
 - k. Display average, minimum, maximum salary of each dept.
 - l. Display dept average of dept whose employee >5.
2. Create following table. Book (id, title, author, publisher, category, year, price)
Distributor(did, name, city, discount) and Order(order_no, title, did, qty)
 - a. Display title and category of all books.
 - b. Display the total no of books per year.
 - c. Display list of authors.
 - d. Display the books published in 1991,92 and 93.
 - e. Display the books published from 1991 to 95.
 - f. Display the books whose price is greater than200.
 - g. Display the total no of books of each category.
 - h. Display titles of all books whose price is greater than average price.
 - i. Display the list of all books whose price is greater then average price of "computer" category.
 - j. Shoe the name of all the distributors who supply "software testing" books.
 - k. Display the details of all books whose price is greater than the maximum of the category average.

1. Display name of all books who are supplying the books whose author is 'Pressman'.

3. Create the following table & solve given queries.

Table Name : branch

Column_name	Datatype	Constraint	Description
Bno	number(4)	Primary key	Branch number
bname	Varchar2(20)	Not null	
City	Varchar2(15)	Not null	

Table Name : customer

Column_name	Datatype	Constraint	Description
Cust_no	Number(6)	Primary key	
Cust_name	Varchar2(20)	Not null	
City	Varchar2(15)	Not null	

Table Name : deposit

Column_name	Datatype	Constraint	Description
Acc_no	Varchar2(5)	Primary key	Starts from 'D'character
Cust_no	Number(6)	Foreign key	references table 'customer'
Bno	Number(4)	Foreign key	Branch number references from table 'branch'

Amount	Number(9,2)	Not null	Default amount is 500.00
Adate	Date	Not null	Date of money deposited

Table Name : borrow

Column_name	Datatype	Constraint	Description
Loan_no	Number(5)	Primary key	
Cust_no	Number(6)	Foreign key	references table 'customer'
Bno	Number(4)	Foreign key	references from table 'branch'
Amount	Number(9,2)	Not null	Default amount is 500.00

- a) Insert minimum 10 records.
- b) describe tables, which are already created.
- c) Give account number and amount of depositors.
- d) Give names of borrowers.
- e) Give names of customers living in city NAGPUR.
- f) Give names of depositors having amount greater than 4000.
- g) Give name of customer having living city BOMBAY and branch city DELHI.
- h) Give names of customer having the same living city as their branch city.
- i) Give name of customers who are borrowers as well as depositors and having living city NAGPUR.
- j) Give name of customers who are depositors and have the same branch city as that of sunil.
- k) Give names of depositors having the same living city as that of shivani and having deposit amount greater than 200.
- l) Give names of borrowers having deposit amount greater than 1000 and loan amount greater than 2000.

- m) Give names of borrowers having loan amount greater than the loan amount of anil.
 - n) Give loan no and loan amount of borrowers having the same branch as that of depositor sunil.
 - o) Give loan no, loan amount, account no, and deposit amount of customers living in city NAGPUR.
4. Write a block to find maximum number.
 5. Write a block for check given number is even or odd.
 6. Write a procedure for addition of two number.
 7. Write a function which return multiplication of two numbers.
 8. Define cursor for display information of student.
 9. Write a procedure for addition and subtraction of two numbers. (Return result).
 10. Create user A and B. create table student (roll_no, name) by user A. Create trigger for avoid update or delete in table by user B.
 11. Create a package for addition and multiplication of two numbers.
 12. Create trigger for avoiding inserting the records whose address 'solapur' and deleting the records whose address 'satara'.(use any table with address field).
 13. Create package for addition, multiplication.
 14. Create function with cursor.
 15. Create package which contain procedure, function , cursor.

Sample Practical Assignment on Angular JS

1. Write angular js app which display your name, college name, class.
2. Write angular js app which demonstrate one way and two way data binding.
3. Demonstrate ng-cut,ng-copy & ng-paste directive.
4. Demonstrate different directive related to mouse and keyboard events.
5. Demonstrate Conditional Directives.
6. Write angular js app which display list of employees in tabular for having different color for even odd row.
7. Write angular js app for creating custom directive which display current date and current time in elements, attributes, class and comment.
8. Demonstrate all types of Expressions used in angular js.
9. Demonstrate nested controller.
10. Demonstrate multiple controllers.
11. Demonstrate number, currency, Uppercase and Lowercase Filters.
12. Write angular js app for displaying current date in 10 different format using date filter.
13. Write angular js app which sort array object data in ascending and descending by using orderby filter.
14. Write angular js app which demonstrate filter filter and json filter.
15. Write angular js code for Creating custom filters.
16. Demonstrate filter in controller for following filter
 - a. Filter
 - b. Date
 - c. Order by
17. Write angular js app which validate data for following validation
 - a. Required
 - b. Email
 - c. Touched
18. Write angular js app which demonstrate all status properties.
19. Write angular js app for validate CSS class.
20. Write angular js code for custom validation.
21. Write angular js code which demonstrate \$apply and \$watch variable.

22. Write angular js code for all types of CSS animation.
23. Write angular js code for animation using nganimate directive.
24. Design simple Single Page Application.
25. Write angular js code for ngRoute module.

Sample Practical Assignment on PHP

1. Write PHP code to check entered number is Armstrong or Not.
2. Write a menu driven program to perform following operations:
3. Check Number is Palindrome or not.
4. Check Number is Perfect or not.
5. Find face value of Entered number.
6. Check Number is Prime or not.
7. Check Number is Strong or not.
8. Write a PHP code to perform following operations: a) Sort array element b) Find Maximum and Minimum number in array c) Merge two arrays in third array. d) Swap two array elements
9. Write a program to overload the constructor.
10. Write a program which uses the static methods and static variables.
11. Write a program to implement different types of inheritance.
12. Write a program to implement interface.
13. Write a program to demonstrate Abstract classes.
14. Write a program to handle different types of exceptions.
15. Write a program which shows the use of 'final' keyword.
16. Write a program to copy the content of one file into another.
17. Write a program to merge two files into third file.
18. Design a web application to perform following task on employee table. I) Add New II) Save III) Delete IV) Update V) Move First VI) Move Last
19. Design a web application that uses cookies object.
20. Design a web application that uses session object.