

Lab	Type	Practical
1	<p><b>A</b></p> <p><b>A</b></p> <p><b>A</b></p> <p><b>A</b></p> <p><b>A</b></p> <p><b>B</b></p> <p><b>B</b></p> <p><b>C</b></p> <p><b>C</b></p>	<p><b><u>Variables, Data Types, Operators</u></b></p> <ol style="list-style-type: none"> <li>1. Write a program to print your name, address, contact number &amp; city.</li> <li>2. Write a program to get two numbers from user and print those two numbers.</li> <li>3. Write program to prompt a user to input his/her name and country name and then output will be shown as given: Hello &lt;yourname&gt; from country &lt;countryname&gt;</li> <li>4. Write a program to calculate the size of the area in square-feet based on Specified length and width.</li> <li>5. Write a program to calculate area of Square, Rectangle and Circle.</li> <li>6. Write a program to calculate Celsius to Fahrenheit and vice-versa using function.</li> <li>7. Write a program to find out Simple Interest using function. (<math>I = PRN/100</math>)</li> <li>8. Write a program to create a Simple Calculator for two numbers (Addition, Multiplication, Subtraction, Division) [Also using if...else &amp; Switch Case]</li> <li>9. Write a program to Swapping without using third variable.</li> <li>10. Write a program to find maximum numbers from given 3 numbers using ternary operator.</li> </ol>

2		<p><b><u>Class and Object, Constructors, Inheritance</u></b></p> <p><b>A</b> 1. Write a program to create a class named <b>Candidate</b> with ID, Name, Age, Weight and Height as data members &amp; also create a member functions like GetCandidateDetails() and DisplayCandidateDetails().</p> <p><b>A</b> 2. Write a program to create a class <b>Staff</b> having data members as Name, Department, Designation, Experience &amp; Salary. Accept this data for 5 different staffs and display only names &amp; salary of those staff who are HOD.</p> <p><b>A</b> 3. Write a program to Create a class <b>Bank_Account</b> with Account_No, Email, User_Name, Account_Type and Account_Balance as data members. Also create a Member function GetAccountDetails() &amp; DisplayAccountDetails().</p> <p><b>A</b> 4. Write a program with following specifications: Class Name: <b>Student</b> Data Members: Enrollment_No, Student_Name, Semester, CPI and SPI Get Students Details using constructor and DisplayStudentDetails() using member function.</p> <p><b>A</b> 5. Write a program to calculate area of a Rectangle using constructor.</p> <p><b>A</b> 6. Write a program for implementing single inheritance which creates one class <b>Account_Details</b> for getting account information and another class <b>Interest</b> for calculating and displaying total interest from the data inserted from account details.</p> <p><b>B</b> 7. Write a program to Define a class <b>Salary</b> which will contain member variable Basic, TA, DA, HRA. Write a program using Constructor with default values for DA and HRA and calculate the salary of employee.</p> <p><b>B</b> 8. Write a program to Define a class <b>Distance</b> have data members dist1, dist2, dist3. Initialize the two data members using constructor and store their addition in third data member using function and display addition.</p> <p><b>C</b> 9. Create a class <b>Furniture</b> with material ,price as data members. Create another class <b>Table</b> with Height , surface_area as data members. Write a program to implement single inheritance.</p> <p><b>C</b> 10. Program to implement the following multiple inheritance using interface</p> <table border="1" data-bbox="395 1630 1444 1760"> <tr> <td>Interface: Gross Method- Gross_sal()</td> <td>Class : Salary Data Members - HRA, TA,DA Methods - Disp_sal()</td> <td>Class : Employee Data Members - Name Methods - basic_sal()</td> </tr> </table>	Interface: Gross Method- Gross_sal()	Class : Salary Data Members - HRA, TA,DA Methods - Disp_sal()	Class : Employee Data Members - Name Methods - basic_sal()
Interface: Gross Method- Gross_sal()	Class : Salary Data Members - HRA, TA,DA Methods - Disp_sal()	Class : Employee Data Members - Name Methods - basic_sal()			

3		<p><b><u>Exception Handling, Interface, Abstraction, String Functions</u></b></p> <div> <div>A</div> <div>1. Write a program to Create a divide by zero exception and handle it.</div> <div>A</div> <div>2. Write a program that reads 5 numbers from user. Demonstrate concept of IndexOutOfRangeException Exception.</div> <div>A</div> <div>3. Write a program to create an <b>abstract class Sum</b> having abstract methods SumOfTwo(int a, int b) and SumOfThree(int a, int b,int c). Create another class <b>Calculate</b> which extends the abstract class and implements the abstract methods.</div> <div>A</div> <div>4. Write a program to create <b>interface Calculate</b>. In this interface we have two member functions Addition() and Subtraction(). Implements this interface in another class named Result.</div> <div>A</div> <div>5. Write program showing use of common methods of String class.</div> <div>B</div> <div>6. Write a program to Replace lower case characters to upper case and Vice-versa.</div> <div>B</div> <div>7. Write a program to create interface named <b>Shape</b>. In this interface, we have three methods Circle(), Triangle() and Square() which calculates area of Circle, Triangle and Square respectively. Implement Shape interface.</div> <div>B</div> <div>8. Write a program to accept a number from the user and throw an exception if the number is not an even number.</div> <div>C</div> <div>9. Write a program to find the longest word in a string.</div> <div>C</div> <div>10. Write a program to change the case of entered character.</div> </div>
4		<p><b><u>Method Overloading, Method Overriding</u></b></p> <div> <div>A</div> <div>1. Write a program using method overloading by changing datatype of arguments to perform addition of two integer numbers and two float numbers.</div> <div>A</div> <div>2. Write a program using method overloading by changing number of arguments to calculate area of square and rectangle.</div> <div>A</div> <div>3. Create a class named RBI with calculateInterest() method. Create another classes HDFC, SBI, ICICI which overrides calculateInterest() method.</div> <div>A</div> <div>4. Create a class Hospital with HospitalDetails() method. Create another classes Apollo, Wockhardt, Gokul_Superspeciality which overrides HospitalDetails() method.</div> <div>B</div> <div>5. Write a programs to Find Area of Square, Rectangle and Circle using Method Overloading.</div> <div>C</div> <div>6. Create a BankAccount class having constructor accepting initialBalance and accountHolderName. Also create Deposit() and withdraw() overloaded methods by which user can deposit/withdraw amount using different types of parameters (ex. cash, check).</div> </div>

5		<p><b><u>Collection Classes</u></b></p> <p><b>A</b></p> <ol style="list-style-type: none"> <li>1. Create an ArrayList for StudentName and perform following operations:             <ol style="list-style-type: none"> <li>a. Add() - To Add new student in list</li> <li>b. Remove() - To Remove Student with specified index</li> <li>c. RemoveRange() - To Remove student with specified range.</li> <li>d. Clear() - To clear all the student from the list</li> </ol> </li> </ol> <p><b>A</b></p> <ol style="list-style-type: none"> <li>2. Create a List for StudentName and perform following operations:             <ol style="list-style-type: none"> <li>a. Add() - To Add new student in list</li> <li>b. Remove() - To Remove Student with specified index</li> <li>c. RemoveRange() - To Remove student with specified range.</li> <li>d. Clear() - To clear all the student from the list</li> </ol> </li> </ol> <p><b>B</b></p> <ol style="list-style-type: none"> <li>3. Create a Stack which takes integer values and perform following operations:             <ol style="list-style-type: none"> <li>a. Push() - To Add new item in stack</li> <li>b. Pop() - To Remove item from the stack</li> <li>c. Peek() – To Return the top item from the stack.</li> <li>d. Contains() - To Checks whether an item exists in the stack or not.</li> <li>e. Clear() - To clear items from stack</li> </ol> </li> </ol> <p><b>B</b></p> <ol style="list-style-type: none"> <li>4. Create a Queue which takes integer values and perform following operations:             <ol style="list-style-type: none"> <li>a. Enqueue() - Adds an item into the queue.</li> <li>b. Dequeue() - Returns an item from the beginning of the queue and removes it from the queue.</li> <li>c. Peek() - Returns an first item from the queue without removing it.</li> <li>d. Contains() - Checks whether an item is in the queue or not</li> <li>e. Clear() - Removes all the items from the queue</li> </ol> </li> </ol> <p><b>C</b></p> <ol style="list-style-type: none"> <li>5. Create a Dictionary collection class object and preform following operations:             <ol style="list-style-type: none"> <li>a. Add: Adds a key-value pair.</li> <li>b. Remove: Removes a key-value pair by key.</li> <li>c. ContainsKey: Checks if a key exists in the hashtable.</li> <li>d. ContainsValue: Checks if a value exists in the hashtable.</li> <li>e. Clear: Removes all key-value pairs.</li> </ol> </li> </ol> <p><b>C</b></p> <ol style="list-style-type: none"> <li>6. Create a Hashtable collection class object and preform following operations:             <ol style="list-style-type: none"> <li>a. Add: Adds a key-value pair.</li> <li>b. Remove: Removes a key-value pair by key.</li> <li>c. ContainsKey: Checks if a key exists in the hashtable.</li> <li>d. ContainsValue: Checks if a value exists in the hashtable.</li> <li>e. Clear: Removes all key-value pairs.</li> </ol> </li> </ol>
---	--	--

6	<p><b>A</b></p> <p><b>A</b></p> <p><b>B</b></p>	<p><b>MVC Overview with Visual Studio</b></p> <p>Introduction to IDE, how to create project of .net core, how to add controllers, action methods and views. How to add NuGet package references.</p> <p>Create a project and add Home Controller with Home, About and Contact Us Action methods with Views. Add appropriate navigation between these pages.</p> <p>Add Admin, User Areas and Define Appropriate Area Routes and Linking Between Areas. Add Home, About and Contact Us views inside User Area.</p>
7	<p><b>A</b></p> <p><b>A</b></p> <p><b>B</b></p>	<p><b>Razor Syntax Overview</b></p> <p>Print table of 5 using Razor</p> <p>Prepare a page which displays student details and his/her semester wise SPI in table format.</p> <p>Prepare semester wise SPI table data in controller file and store it in ViewBag. Display the data to view page using foreach loop.</p>
8	<p><b>A</b></p> <p><b>B</b></p>	<p><b>Design a Static Web using Bootstrap</b></p> <p>Create a project and add Home, About, Contact Us views. And add appropriate routing between these pages. Use bootstrap for better design. (use conventional routing)</p> <p>Add attribute routing in above practical along with optional parameters and apply ignore route templates functionality.</p>
9	<p><b>A</b></p> <p><b>A</b></p>	<p><b>Static CRUD</b></p> <p>Prepare a static CRUD for employee table/model which displays employee details in tabular format. Create employee model class for it and use List collection class object to pass data from controller to view.</p> <p>Do create contact us form &amp; display entered data below in same view page.</p>
10	<p><b>A</b></p> <p><b>B</b></p>	<p><b>Theme Conversion</b></p> <p>Single page bootstrap theme conversion [Personal CV].</p> <p>Multiple page admin theme conversion for the project with required pages.</p>
11	<p><b>A</b></p> <p><b>B</b></p> <p><b>B</b></p> <p><b>C</b></p>	<p><b>Create Database and prepare stored procedures for Select command.</b></p> <p>Create Database : AddressBook also create SelectAll and SelectByPK stored procedure for Country, State and City tables.</p> <p>Pass the City Name in procedure and display all the records based on city name.</p> <p>Pass the State Name in procedure and display all the cities belongs to the that state.</p> <p>Pass the Country name in procedure and display all the states with number of cities.</p>

12	A	<b>Prepare stored procedure for Insert, Update and Delete command</b> Create all tables Insert, Update and Delete stored procedures for Country, State and City tables.
	B	Create an Insert, Update & Delete Stored Procedure for your own one table with minimum 6-7 columns.
13	A	<b>Implementation of Html Helpers</b> Student registration form using Standard html helpers. (StudentName, Branch, Semester, Birthdate, Mobile, Email, Address, City, Hobbies, Gender)
	A	Student registration form using Strongly typed html helpers. (StudentName, Branch, Semester, Birthdate, Mobile, Email, Address, City, Hobbies, Gender)
	B	Employee Registration form using Standard html helpers.
	C	Job Inquiry form using Strongly typed html helpers.
14	A	<b>Demonstration of File Upload</b> Design a view by which user can upload his/her resume to the server and display the uploaded resume.
	B	Design a view from user can upload their profile picture.
15	A	<b>Project Creation with Areas</b> Create a new asp.net core project with MVC Template and Create appropriate MVC Areas for AddressBook Project.
16	A	<b>Prepare Design Pages</b> Design List Page & Add/Edit Pages.
17	A	<b>Routing</b> Apply Attribute Routing between all the pages. Use appropriate routing attributes as required.
18	A	<b>Model creation and Data annotation</b> Prepare model classes as per requirement and Implement data annotation on all the model classes.
19	A	<b>Database connectivity and Implementation of Read operation</b> Create database connectivity and Display data (All Records) for Country List view page.
	B	Display the data for State List and City List view pages.

20	A B	<b>Implementation of Delete functionality</b> Implement Delete functionality for LOC_Country, LOC_State, LOC_City. Add prompt with 'Are you sure you want to delete record?' using Sweet alert when user want to delete any record.
21	A B	<b>Apply Server side validation</b> Implement insert functionality for Country view page. Add dropdown for Country, State as required and implement insert functionality for State, City view pages with required validations.
22	A B	<b>Implementation of Insert functionality</b> Implement insert functionality for Country view page. Add dropdown for Country, State as required and implement insert functionality for State, City view pages with required validations.
23	A	<b>Implementation of Update functionality</b> Implement update functionality for Country, State & City view pages with appropriate message as required.
24	A B	<b>Login and User Registration operation</b> Implement Login functionality. Implement User Registration functionality.
25	A B	<b>Implementation of Excel Export functionality</b> Implement Search functionality for all the list pages. Add a button by which user can export table data to excel.
26	A	<b>URL Encryption-Decryption</b> Perform URL Encryption and Decryption using standard encryption decryption algorithm in whole project.