Lab	Type	Practical
1		**Variables, Data Types, Operators**
	A	1. Write a program to print your name, address, contact number & city.
	A	2. Write a program to get two numbers from user and print those two numbers.
	A	3. Write program to prompt a user to input his/her name and country name and then output will be shown as given: Hello <yourname> from country <countryname>
	A	4. Write a program to calculate the size of the area in square-feet based on Specified length and width.
	A	5. Write a program to calculate area of Square, Rectangle and Circle.
	B	6. Write a program to calculate Celsius to Fahrenheit and vice-versa using function.
	B	7. Write a program to find out Simple Interest using function. (I = PRN/100)
	B	8. Write a program to create a Simple Calculator for two numbers (Addition, Multiplication, Subtraction, Division) [Also using if…else & Switch Case]
	C	9. Write a program to Swapping without using third variable.
	C	10. Write a program to find maximum numbers from given 3 numbers using ternary operator.

2		**Class and Object, Constructors**
	A	1. Write a program to create a class named **Candidate** with ID, Name, Age, Weight and Height as data members & also create a member functions like GetCandidateDetails() and DisplayCandidateDetails().
	A	2. Write a program to create a class **Staff** having data members as Name, Department, Designation, Experience & Salary. Accept this data for 5 different staffs and display only names & salary of those staff who are HOD.
	A	3. Write a pogram to Create a class **Bank_Account** with Account_No, Email, User_Name, Account_Type and Account_Balance as data members. Also create a Member function GetAccountDetails() & DisplayAccountDetails().
	A	4. Write a program with following specifications: Class Name: **Student** Data Members: Enrollment_No, Student_Name, Semester, CPI and SPI Get Students Details using constructor and DisplayStudentDetails() using member function.
	B	5. Write a program to calculate area of a Rectangle using constructor.
	B	6. Write a program for implementing single inheritance which creates one class **Account_Details** for getting account information and another class **Interest** for calculating and displaying total interest from the data inserted from account details.
	C	7. Write a program to Define a class **Salary** 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.
	C	8. Write a program to Define a class **Distance** 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.
3		**Method Overloading, Method Overriding**
	A	1. Write a program using method overloading by changing datatype of arguments to perform addition of two integer numbers and two float numbers.
	A	2. Write a program using method overloading by changing number of arguments to calculate area of square and rectangle.
	A	3. Create a class named RBI with calculateInterest() method. Create another classes HDFC, SBI, ICICI which overrides calculateInterest() method.
	A	4. Create a class Hospital with HospitalDetails() method. Create another classes Apollo, Wockhardt, Gokul_Superspeciality which overrides HospitalDetails() method.
	B	5. Write a programs to Find Area of Square, Rectangle and Circle using Method Overloading.
	C	6. Create a BankAccount class having constructor accepting initialBalance and accountHolderName. Also create Deposite() and withdraw() overloaded methods by which user can deposit/withdraw amount using different types of parameters (ex. cash, check).

4 **Inheritance & String Functions**

A
1. Write program showing use of common methods of String class.

A
2. Write a program to Replace lower case characters to upper case and Vice-versa.

A
3. Write a program to find the longest word in a string.

A
4. Write a program to change the case of entered character.

B
5. Create a base class Person with the Data Members string name, int age and member functions void AcceptDetails(string name, int age) – to accept values, void DisplayDetails() – to display value. Create a derived class Student that inherits from Person, with additional members int rollNo, double marks and member functions void AcceptStudentDetails(int rollNo, float marks) – to accept student info, void DisplayStudentDetails() – to display student info along with base class data.

B
6. Create a base class Device with the data members string deviceId, string manufacturer and member functions - void SetDevice(string id, string mfg), void ShowDevice(),Derive class SmartDevice from Device with Data Members string os, int ram and member functions void SetSmartDevice(string os, int ram), void ShowSmartDevice(), Further derive class Smartphone from SmartDevice having data members int cameraMP, int batteryMah and member functions void SetSmartphone(int camera, int battery), void ShowSmartphone() – displays complete info from all levels.

C
7. Create a class **Furniture** with material ,price as data members. Create another class **Table** with Height , surface_area as data members. Write a program to implement single inheritance.

C
8. Program to implement the following multiple inheritance using interface

Interface: Gross Method- Gross_sal()	Class : Salary Data Members - HRA, TA,DA Methods - Disp_sal()	Class : Employee Data Members - Name Methods - basic_sal()

5		**Exception Handling, Interface, Abstraction**
	A	1. Write a program that reads two integers from the user. Perform division and demonstrate the concept of DivideByZeroException using try-catch block to handle invalid division scenarios.
	A	2. Write a program that reads 5 numbers from user. Demonstrate concept of IndexOutOfRange Exception.
	A	3. Write a program that reads a number from the user and converts it using int.Parse(). If the input is non-numeric (like alphabets), demonstrate the concept of FormatException using appropriate exception handling.
	A	4. Write a program to create an **abstract class Sum** having abstract methods SumOfTwo(int a, int b) and SumOfThree(int a, int b,int c). Create another class **Calculate** which extends the abstract class and implements the abstract methods.
	B	5. Write a program to create **interface Calculate**. In this interface we have two member functions Addition() and Subtraction(). Implements this interface in another class named Result.
	B	6. Write a program to create interface named **Shape**. In this interface, we have three methods Circle(), Triangle() and Square() which calculates area of Circle, Triangle and Square respectively. Implement Shape interface.
	C	7. Create an interface IPayable with a method double CalculateSalary(). Create a class Employee with Data Members: string name, double baseSalary, double bonus and Member function(s) - CalculateSalary() - returns base salary + bonus. Demonstrate the use of interface to calculate salary of different employees.
	C	8. Write a program to accept a number from the user and throw an exception if the number is not an even number.

6		**Collection Classes**
	A	1. Create an ArrayList for StudentName and perform following operations: a. Add() - To Add new student in list b. Remove() - To Remove Student with specified index c. RemoveRange() - To Remove student with specified range. d. Clear() - To clear all the student from the list
	A	2. Create a List for StudentName and perform following operations: a. Add() - To Add new student in list b. Remove() - To Remove Student with specified index c. RemoveRange() - To Remove student with specified range. d. Clear() - To clear all the student from the list
	B	3. Create a Stack which takes integer values and perform following operations: a. Push() - To Add new item in stack b. Pop() - To Remove item from the stack c. Peek() – To Return the top item from the stack. d. Contains() - To Checks whether an item exists in the stack or not. e. Clear() - To clear items from stack
	B	4. Create a Queue which takes integer values and perform following operations: a. Enqueue() - Adds an item into the queue. b. Dequeue() - Returns an item from the beginning of the queue and removes it from the queue. c. Peek() - Returns a first item from the queue without removing it. d. Contains() - Checks whether an item is in the queue or not e. Clear() - Removes all the items from the queue
	C	5. Create a Dictionary collection class object and preform following operations: a. Add: Adds a key-value pair. b. Remove: Removes a key-value pair by key. c. ContainsKey: Checks if a key exists in the hashtable. d. ContainsValue: Checks if a value exists in the hashtable. e. Clear: Removes all key-value pairs.
	C	6. Create a Hashtable collection class object and preform following operations: a. Add: Adds a key-value pair. b. Remove: Removes a key-value pair by key. c. ContainsKey: Checks if a key exists in the hashtable. d. ContainsValue: Checks if a value exists in the hashtable. e. Clear: Removes all key-value pairs.
7	**A**	**MVC Overview with Visual Studio** Introduction to IDE, how to create project of .net core, how to add controllers, action methods and views. How to add NuGet package references.
	A	Create a project and add Home Controller with Home, About and Contact Us Action methods with Views. Add appropriate navigation between these pages.

8	**A** **A** **B** **C**	**Static CRUD** Prepare employee page which displays employee details in table format. Create employee model class for it and use List collection class object to pass data from controller to view. Add delete functionality in table page. Add functionality to insert a record. Add functionality to update a record.
9	**A**	**Design a Static Web using Bootstrap** Create a project and add Home, About, Contact Us views. And add appropriate routing between these pages. Use bootstrap for better design.
10	**A**	**Create Database and prepare stored procedures for Select command** Create Database : StudentMaster also Create all tables SelectAll and SelectByPK stored procedures
11	**A**	**Prepare stored procedure for Insert, Update and Delete command** Create all tables Insert, Update and Delete stored procedures
12	**A** **B**	**Theme Conversion** Single page bootstrap theme conversion [Personal CV]. Multiple page admin theme conversion for the project with required pages.
13	**A** **B**	**Demonstration of File Upload** Design a view by which user can upload his/her resume to the server and display the uploaded resume. Design a view from user can upload their profile picture.
14	**A** **A** **B** **C**	**Implementation of Html Helpers** Student registration form using Standard html helpers. (StudentName, Branch, Semester, Birthdate, Mobile, Email, Address, City, Hobbies, Gender) Student registration form using Strongly typed html helpers. (StudentName, Branch, Semester, Birthdate, Mobile, Email, Address, City, Hobbies, Gender) Employee Registration form using Standard html helpers. Job Inquiry form using Strongly typed html helpers.
15	**A**	**Project Creation** Create a new asp.net core project with MVC Template and Create appropriate MVC Areas for Country, State, City, Branch, and Student
16	**A**	**Prepare Design Pages** Design List Page & Add/Edit Pages. [For LOC_Country, LOC_State, LOC_City]
17	**A**	**Routing** Apply Attribute Routing in whole Project

18	**A**	**Model creation and Data annotation** Implement data annotation on all the model classes.
19	**A**	**Database connectivity and Implementation of read operation** Create Database connectivity and Display data (All Records) for LOC_CountryList.cshtml, LOC_StateList.cshtml, LOC_CityList.cshtml view pages.
20	**A**	**Apply Server side validation** Apply server side validations with proper message.
21	**A**	**Implementation of Delete functionality** Implement Delete functionality for LOC_Country, LOC_State, LOC_City with prompt Are you sure you want to delete record?
22	**A**	**Implementation of Insert functionality** Implement Insert functionality for LOC_Country, LOC_State, LOC_City view pages with required validations
23	**A**	**Implementation of Update functionality** Implement Update functionality for LOC_Country, LOC_State, LOC_City view pages.
24	**A** **B**	**Login and User Registration operation** Implement Login functionality. Implement User Registration functionality.
25	**A** **B**	**Implementation of Search functionality and Excel Export functionality** Implement Search functionality for all the list pages to demonstrate IFormCollection class Add a button by which user can export table data to excel.
26	**A**	**Implementation of Cascade dropdown functionality** Implement functionality to fill state wise city using cascade dropdown functionality.
27	**A**	**URL Encryption-Decryption** Perform URL Encryption and Decryption using standard encryption decryption algorithm in whole project.
28	**A**	**Prepare Web API to apply HTTP Get and Delete Methods** Create an in-memory student list having Id, Name, Age and Email. Implement GET, DELETE endpoints and return appropriate status codes.
29	**A**	**Prepare Web API to apply HTTP Post and Put Methods** Create an in-memory student list having Id, Name, Age and Email. Implement POST, PUT endpoints and return appropriate status codes.

30		**Implement JSON Serialization and Deserialization**
	A	Create an Employee class with properties EmployeeId, FirstName, LastName, Email, PhoneNumber, IsActive and DateOfBirth.
	A	Use JsonIgnore to prevent the PhoneNumber from being included in the serialized JSON.
	B	Creating methods and returning response with status codes and http verbs.