.NET TECHNOLOGIES (01CE0523)
Lab Manual

Name: Yug Mehta

Enrolment No: 92301703058

Class: 5EC4

Batch:A

INDEX

Lab	Program	Date	Marks	Signature
1.	Write a program to create an Employee class with variables EmpId, EmpName, Department, and Salary. Include a constructor to initialize these variables, create objects of the class, and display employee details using a display method.			
2.	Create a menu-driven program with the following cases: Case 1: AIM: Write a program to implement single inheritance with the base class Name: Furniture and Derived Class name: Chair. Accept and display data for one Table. Case 2: Write a program to implement multilevel inheritance with classes Student (roll_no, name), Test (marks1, marks2), and Result (total), where each derived class adds its own data members. Case 3: Write a program for the above class hierarchy for the Employee, where the base class is Employee, and the derived classes are Programmer and Manager. Case 4: Write a C# program to demonstrate interfaces and polymorphism by creating a Vehicle interface with a speedUp method, implementing it in Bicycle and Bike classes that track and display speed.			

3.	Create a menu-driven program with the following cases: Case 1: Write a program to demonstrate method overloading by creating multiple methods with the same name to perform addition and subtraction operations using different parameter types. Case 2: Write a program that uses method overriding to calculate the surface area of a rectangle and a square, with base and child classes validating dimensions and handling different shape calculations. Case 3: Write a program in C# to demonstrate a Number DivideByZero Exception handle the exception using trycatch.			

4.	Create a web form using ASP.NET Web Forms to collect student details like Name, Email, Job Position, Position (as a dropdown), Salary, and File Upload. Add a checkbox for agreement and use required.			
5.	Write the server-side code to handle the above form submission. Verify that the uploaded file is in PDF or DOC format. Make sure the agreement checkbox is checked.			

6.	In the above web application, implement appropriate fieldwise validation for all input fields. Key Validations Implemented: 1. Personal Details Validation: - Name: Required field - Email: Required + regex validation - Phone: Required + 10-digit regex validation - DOB: Required + date range validation - Gender: Radio button selection required 2. Academic Information Validation: - Course: Dropdown selection required - Qualification: Dropdown selection required - Institution: Required text field 3. Skills Validation: - Required multiline text field 4. Additional Features: - ValidationSummary for consolidated error display. Implementation Notes: 1. Required Fields: All mandatory fields use RequiredFieldValidator 2. Format Validation: - Email uses RegularExpressionValidator with a standard email pattern - Phone uses regex for 10-digit validation - DOB uses RangeValidator with date constraints			
7.	a) Create a web application using ASP.NET Web Forms with fields for username and password using session management to track the logged-in user. b) Create a web application using ASP.NET Web Forms that allows users to create a cookie with their information and retrieve the stored cookie data for			

	display. Fields are such as Username & Password.			
8.	An e-commerce database stores transactional data related to online shopping, such as customers, orders, and shopping carts. The Shopping table in the database typically includes the following fields: • CartID (Primary Key): Unique identifier for the cart. • ProductName: Identifier for the product. • CustomerName: Identifier for the customer. • Quantity: Number of products in the cart. • AddedDate: Date the product was added to the cart. Status: Status of the cart (e.g., Active, Purchased). Using ADO.NET, CRUD (Create, Read, Update, Delete) operations are performed on the Shopping table, allowing for seamless interactions with the database. ADO.NET components, such as SqlConnection, SqlCommand, SqlDataReader, and SqlDataAdapter, are used to execute SQL queries for managing shopping cart data.			
9.	ASP.NET project using MVC.			
10.	Create a Web API using ASP.NET to build HTTP-based services that enable applications to communicate. For example, in a library system, it allows users to fetch, add, update, or delete book records using methods such as GET, POST, PUT, and DELETE.			
11.	Implement security in ASP.NET Web API to ensure that only authorized users can access resources. For example, in a library system, only administrators can add or delete books using authentication methods such as forms or Windows login, while filters and routing control who can access which endpoints.			

Experiment 1

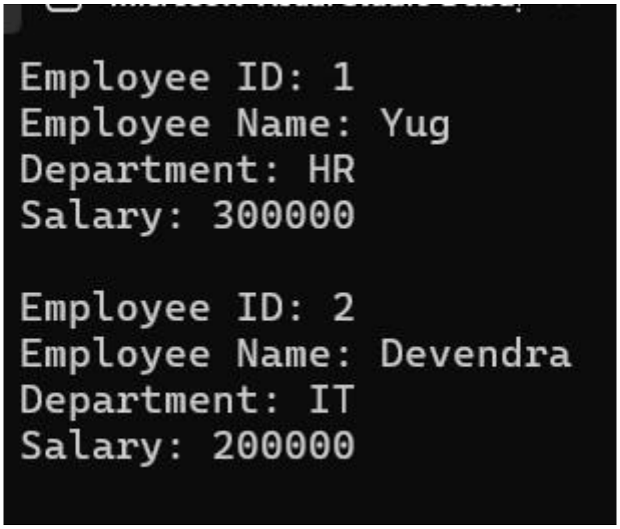
AIM: Write a program to create an Employee class with variables EmpId, EmpName, Department, and Salary, include a constructor to initialize these variables, create objects of the class, and display employee details using a display method.

Source: Program.cs

Code:

```
using System;

 class Employee

{

 public int EmpId;
 public string EmpName;

public string Department;

public double Salary;


    public Employee(int empId, string empName, string department, double
salary)


    {

        EmpId = empId;

        EmpName = empName;

        Department = department;

        Salary = salary;

    }


     public void

Display()

    {

        Console.WriteLine("Employee ID: " + EmpId);
        Console.WriteLine("Employee Name: " + EmpName);

        Console.WriteLine("Department: " + Department);

        Console.WriteLine("Salary: " + Salary);

        Console.WriteLine();

    }

}

  Class

Program

{   public static void Main()

    {


        Employee emp1 = new Employee(1, "Yug", "HR", 300000);

        Employee emp2 = new Employee(2, "Devendra", "IT", 200000);


emp1.Display();

emp2.Display();

    }

}
```

Output: Screenshot:



```
Employee ID: 1
Employee Name: Yug
Department: HR
Salary: 300000

Employee ID: 2
Employee Name: Devendra
Department: IT
Salary: 200000
```