C# and .NET Frameworks

Assignment 1

1.Develop the C# program to initialize two dimensional array and print all the elements of the array on the same line separated with space.

AIM:

To initialize and print a 2D array containing numbers 1 to 9.

PROGRAM:

```
using System;

class Program
{
    static void Main()
    {
        // Initialize a 2D array
        int[,] array = {
            { 1, 2, 3 },
            { 4, 5, 6 },
            { 7, 8, 9 }
        };

        // Get the dimensions of the array
        int rows = array.GetLength(0);
        int columns = array.GetLength(1);
```
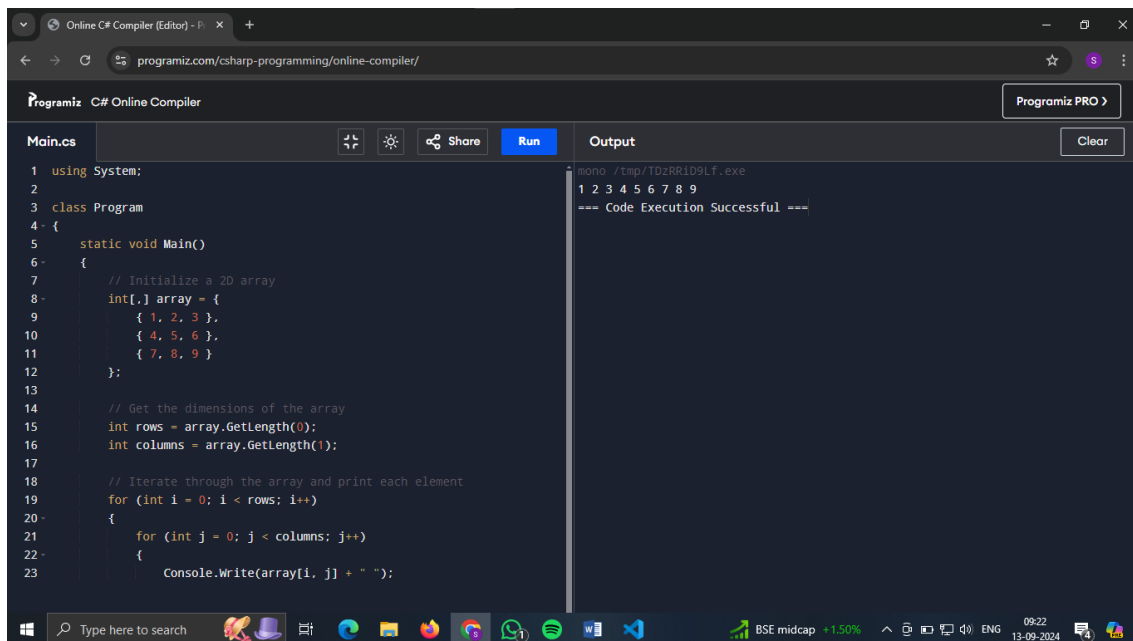
```
    // Iterate through the array and print each element

    for (int i = 0; i < rows; i++)

    {

        for (int j = 0; j < columns; j++)

        {

            Console.Write(array[i, j] + " ");

        }

    }

    }

}
```


OUTPUT:

123456789

2. Aravind wants to apply for competitive exam. He needs to know whether he is eligible to apply. The eligibility criteria is given below:

- **Age should be greater than 18 years, but not more than 30.**
- **The candidate should have passed 10 std with a minimum pass percentage of 65.**

Design the C# program to help him to know his eligibility. If the criteria gets satisfied, print he is eligible else print he is not eligible

AIM:

To determine and print whether a person named Aravind is eligible to apply for a competitive exam based on their age and 10th standard pass percentage.

PROGRAM:

```
using System;

class Program
{
    static void Main()
    {
        // Input details for Aravind
        Console.Write("Enter Aravind's age: ");
        int age = int.Parse(Console.ReadLine());

        Console.Write("Enter Aravind's 10th standard pass percentage: ");
        double passPercentage = double.Parse(Console.ReadLine());

        // Check eligibility criteria
        bool isEligible = (age > 18 && age <= 30) && (passPercentage >= 65);

        // Print eligibility status
        if (isEligible)
        {
            Console.WriteLine("Aravind is eligible to apply for the competitive exam.");
```
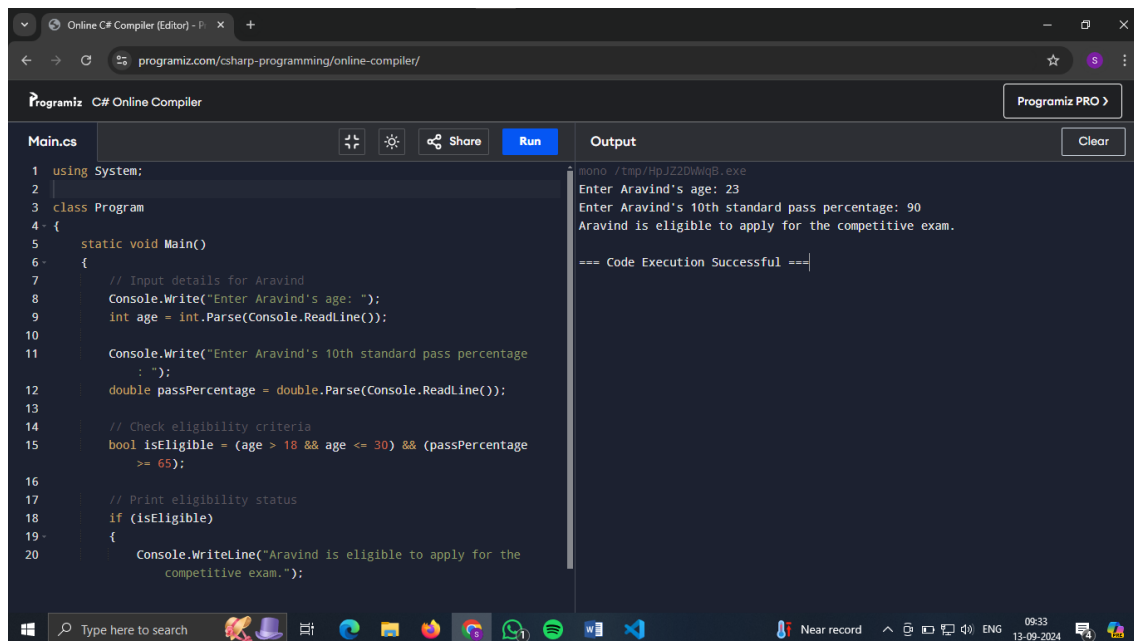
```
        }
        else
        {
            Console.WriteLine("Aravind is not eligible to apply for the competitive
exam.");
        }
    }
}
```


INPUT:

Enter Aravind's age: 23

Enter Aravind's 10th standard pass percentage: 90

OUTPUT:

Aravind is eligible to apply for the competitive exam.

3. Design the C# console application named validation to get mobile number as input from the user. Validate the mobile number with the following cases:

- **The first four number must be followed by then followed by next six numbers(eg:9894-256874)**
- **Should contains only numbers**
- **Should be of length 10.**
- **The first number should start only with 9 Or 8**

AIM

To validate and print whether a given mobile number is valid or not.

PROGRAM:

```
using System;
using System.Text.RegularExpressions;

class Validation
{
  static void Main()
  {
    // Prompt the user to enter the mobile number
    Console.Write("Enter the mobile number: ");
    string mobileNumber = Console.ReadLine();

    // Validate the mobile number
    if (IsValidMobileNumber(mobileNumber))
    {
      Console.WriteLine("The mobile number is valid.");
    }
```

```csharp
        else
        {
            Console.WriteLine("The mobile number is invalid.");
        }
    }


    static bool IsValidMobileNumber(string mobileNumber)
    {
        // Check the format using a regular expression
        // Pattern explanation:
        // ^         : Start of string
        // [98]      : The first digit should be 9 or 8
        // [0-9]{3}  : The next three digits should be numbers
        // -         : A hyphen
        // [0-9]{6}  : The next six digits should be numbers
        // $         : End of string
        string pattern = @"^[98][0-9]{3}-[0-9]{6}$";
        if (Regex.IsMatch(mobileNumber, pattern))
        {
            return true;
        }
        return false;
    }
}
```

INPUT:

1234567890

OUTPUT:

The mobile number is invalid.

4. Write the missing code snippets and the statements in the C# program given below.

```
Class person {
_____name;
_____age;
_____weight;
Void printperson() {
// write the code to print name, age and weight of a person
}
}
Class persondata {
Static void Main(string[] args) {
```

```
    person_____ = _____;
     _____.name = "Kannan";
    _____.age = 19;
    _____.weight = 58;
    // write the statement to access printperson() function
    }
    }
```

AIM:

   To create a Person class, instantiate it, and print out the person's name, age, and weight using a method.

PROGRAM:
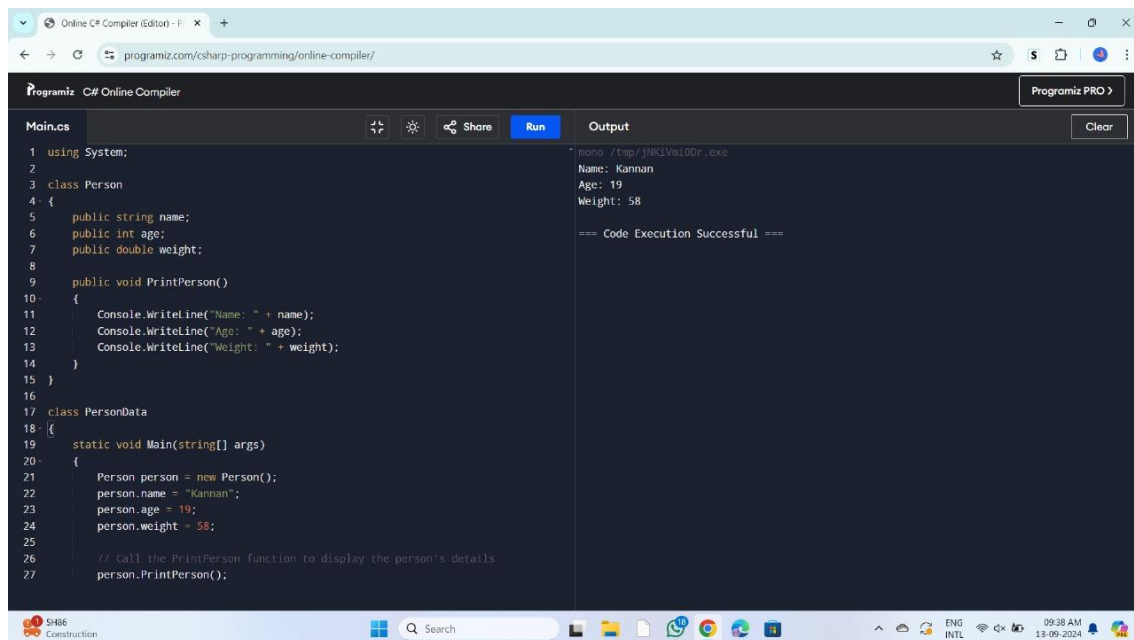
```csharp
using System;

class Person
{
  public string name;
  public int age;
  public double weight;

  public void PrintPerson()
  {
    Console.WriteLine("Name: " + name);
    Console.WriteLine("Age: " + age);
    Console.WriteLine("Weight: " + weight);
  }
}
```

```
class PersonData
{
    static void Main(string[] args)
    {
        Person person = new Person();
        person.name = "Kannan";
        person.age = 19;
        person.weight = 58;


        person.PrintPerson();
    }
}
```


OUTPUT:

Name: Kannan

Age: 19

Weight: 58

5. A hospital wants to create a console application to maintain its impatient details. The information to store includes:

- **Name of the patient**
- **Date of admission**
- **Age of patient**
- **Disease**
- **Date of discharge**
- **Total bills paid**

Design the C# program with the class name patient with necessary data members to store the above information. The class should have two member functions, one to get the patients information and other to display the information. Create a main class called hospital to create necessary instances, methods calling statements and display all the details about the patient.

AIM:

To create a Patient class, collect patient information through user input, and display the collected information using methods.

PROGRAM:

uaing System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Assignment1_Q5

{

    class Patient

```csharp
{
    // Data members to store patient information
    private string name;
    private DateTime dateOfAdmission;
    private int age;
    private string disease;
    private DateTime dateOfDischarge;
    private decimal totalBillsPaid;

    // Method to get patient information from the user
    public void GetPatientInfo()
    {
        Console.Write("Enter Patient Name: ");
        name = Console.ReadLine();

        Console.Write("Enter Date of Admission (yyyy-mm-dd): ");
        dateOfAdmission = DateTime.Parse(Console.ReadLine());

        Console.Write("Enter Age of Patient: ");
        age = int.Parse(Console.ReadLine());

        Console.Write("Enter Disease: ");
        disease = Console.ReadLine();

        Console.Write("Enter Date of Discharge (yyyy-mm-dd): ");
        dateOfDischarge = DateTime.Parse(Console.ReadLine());
```

```csharp
            Console.Write("Enter Total Bills Paid: ");

            totalBillsPaid = decimal.Parse(Console.ReadLine());

        }


        // Method to display patient information

        public void DisplayPatientInfo()

        {

            Console.WriteLine("\nPatient Details:");

            Console.WriteLine($"Name: {name}");

            Console.WriteLine($"Date of Admission:
{dateOfAdmission.ToShortDateString()}");

            Console.WriteLine($"Age: {age}");

            Console.WriteLine($"Disease: {disease}");

            Console.WriteLine($"Date of Discharge:
{dateOfDischarge.ToShortDateString()}");

            Console.WriteLine($"Total Bills Paid: {totalBillsPaid:C}");

        }

    }

    class Program

    {

        static void Main(string[] args)

        {

            Patient patient = new Patient();


            // Get patient information

            patient.GetPatientInfo();


            // Display patient information
```

```
            patient.DisplayPatientInfo();


            // Wait for user input before closing

            Console.WriteLine("\nPress any key to exit...");

            Console.ReadKey();


        }

    }

}
```


INPUT:

Enter Patient Name:raja

Enter Date of Admission (yyyy-mm-dd): 2024-08-12

Enter Age of Patient: 32

Enter Disease: Fever

Enter Date of Discharge (yyyy-mm-dd): 2024-08-20

Enter Total Bills Paid: 25,000

OUTPUT:

Patient Details:

Name: aja

Date of Admission: 08/12/2024

Age: 32

Disease: Fever

Date of Discharge: 08/20/2024

Total Bills Paid: ?25,000.00

6. **Implement the C# code to get two vector number as input, add them and print the sum as another vector. Make use of operator overloading to perform addition of vector numbers.**

AIM:

To create a Vector class, overload the '+' operator to add two vectors, and demonstrate vector addition by taking user input for two vectors and displaying their sum.

PROGRAM:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Assignment1_Q6
```

```csharp
{
  class Vector
  {
    public int X { get; set; }
    public int Y { get; set; }

    // Constructor to initialize vector components
    public Vector(int x, int y)
    {
      X = x;
      Y = y;
    }

    // Overload the '+' operator to add two vectors
    public static Vector operator +(Vector v1, Vector v2)
    {
      return new Vector(v1.X + v2.X, v1.Y + v2.Y);
    }

    // Method to display the vector

  }
  class Program
  {
    static void Main(string[] args)
    {
      // Input first vector
      Console.WriteLine("Enter the first vector:");
      Console.Write("X1: ");
      int x1 = int.Parse(Console.ReadLine());
      Console.Write("Y1: ");
      int y1 = int.Parse(Console.ReadLine());
      Vector vector1 = new Vector(x1, y1);

      // Input second vector
      Console.WriteLine("Enter the second vector:");
      Console.Write("X2: ");
      int x2 = int.Parse(Console.ReadLine());
      Console.Write("Y2: ");
      int y2 = int.Parse(Console.ReadLine());
```

```
        Vector vector2 = new Vector(x2, y2);

        // Add the vectors using overloaded '+' operator
        Vector sumVector = vector1 + vector2;

        // Display the result
        Console.WriteLine("\nSum of the vectors:");
        Console.WriteLine($"{sumVector.X}  {sumVector.Y}");

        // Wait for user input before closing
        Console.WriteLine("\nPress any key to exit...");
        Console.ReadKey();

    }
  }
}
```
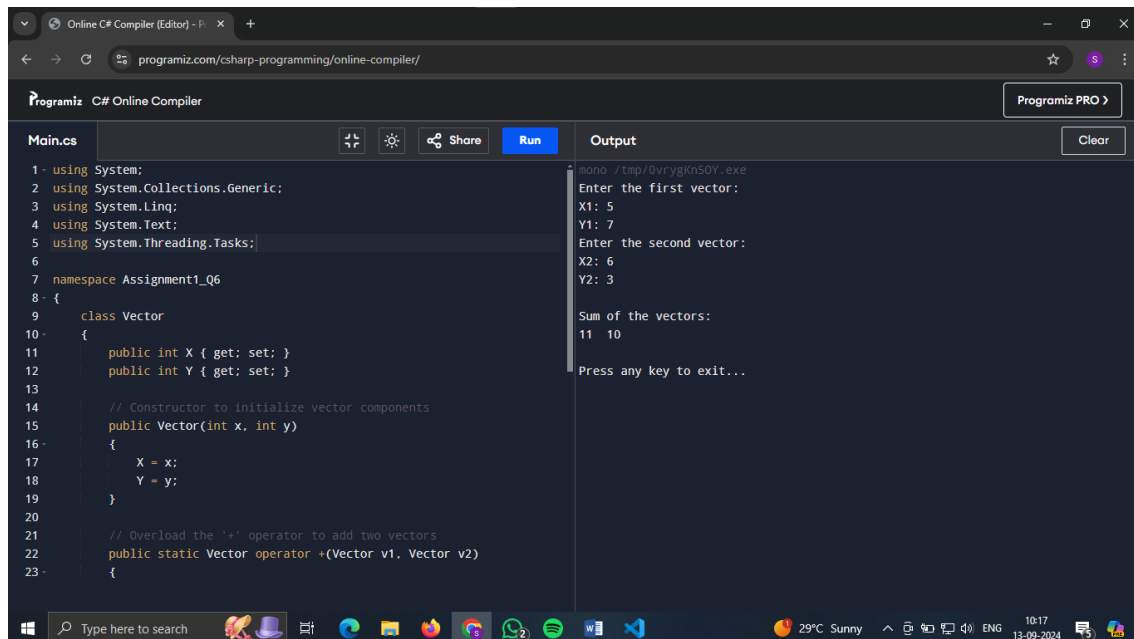

INPUT:
Enter the first vector:
X1: 5

Y1: 7
Enter the second vector:
X2: 6
Y2: 3

OUTPUT:
Sum of the vectors:
    11 10

7. **Create the class student with necessary members to maintain the basic details of a student such as name, age, address and mobile number. Add method getDate() to read the basic details and printData() to print the details of the student. Inherit the student class into the sub class called studentmark with necessary members to maintain student mark details. Override the getData() and printData() in student mark class to read mark details and print the marks, respectively. Also, define a method to find the grade of the student based on his/her marks. Design the student main class to access the member of both the classes in C#**

AIM:

To create a Student class and a derived StudentMark class, which inherits and extends the base class to include mark details, calculates grades based on marks, and demonstrates polymorphism through overridden methods.

PROGRAM:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Assignment1_Q7
{
```

```csharp
class Student
{
    // Data members for basic details
    protected string name;
    protected int age;
    protected string address;
    protected string mobileNumber;

    // Method to get basic details of the student
    public virtual void getData()
    {
        Console.Write("Enter Student Name: ");
        name = Console.ReadLine();

        Console.Write("Enter Age: ");
        age = int.Parse(Console.ReadLine());

        Console.Write("Enter Address: ");
        address = Console.ReadLine();

        Console.Write("Enter Mobile Number: ");
        mobileNumber = Console.ReadLine();
    }

    // Method to print basic details of the student
    public virtual void printData()
    {
        Console.WriteLine("\nStudent Details:");
        Console.WriteLine($"Name: {name}");
        Console.WriteLine($"Age: {age}");
        Console.WriteLine($"Address: {address}");
        Console.WriteLine($"Mobile Number: {mobileNumber}");
    }
}

// Subclass to maintain student mark details
class StudentMark : Student
{
    // Data members for mark details
```

```csharp
        private int marks;

        // Override method to get student marks
        public override void getData()
        {
            // Call base method to get basic details
            base.getData();

            Console.Write("Enter Marks: ");
            marks = int.Parse(Console.ReadLine());
        }

        // Override method to print student marks
        public override void printData()
        {
            // Call base method to print basic details
            base.printData();

            Console.WriteLine($"Marks: {marks}");
            Console.WriteLine($"Grade: {FindGrade()}");
        }

        // Method to determine grade based on marks
        private string FindGrade()
        {
            if (marks >= 90) return "A";
            else if (marks >= 75) return "B";
            else if (marks >= 60) return "C";
            else if (marks >= 50) return "D";
            else return "F";
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            // Create an instance of the StudentMark class
            StudentMark studentMark = new StudentMark();
            studentMark.getData();
```
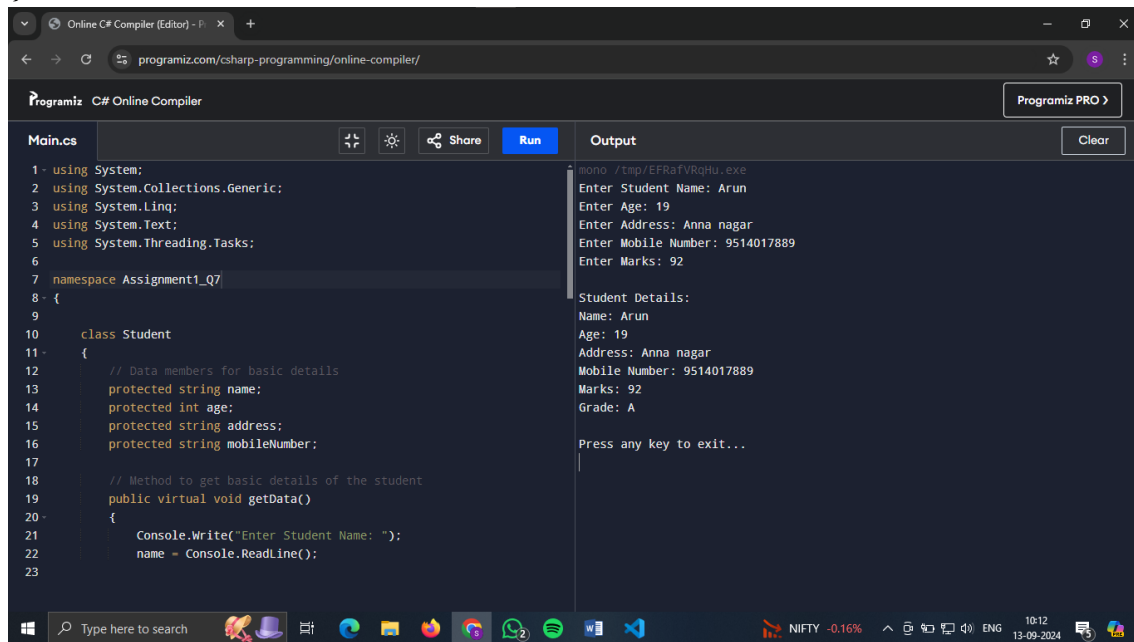
```
            studentMark.printData();

            // Wait for user input before closing
            Console.WriteLine("\nPress any key to exit...");
            Console.ReadKey();
        }
    }
}
```


INPUT:
Enter Student Name: Arun
Enter Age: 19
Enter Address: Anna nagar
Enter Mobile Number: 9514017889
Enter Marks: 92

OUTPUT:
Student Details:
Name: Arun
Age: 19
Address: Anna nagar
Mobile Number: 9514017889
Marks: 92
Grade: A

8. **Design sample C# program with class name employee to compute netsalary of the employee using the basic salary, if for the job_catg is 1 use table-l else use table-II. Use constructor to initialize basic salary,hra,da,pf and loan. The employee class should contain input() method to get input for job_catg, empno, empname, calculateSalary() method to compute salary and display() method to print the details.**

Table-I	Table-II
BASIC=Rs. 8,000 HRA=10% of basic DA=20% of basic LOAN=Rs. 300 PF=Rs. 500	BASIC=Rs. 15,000 HRA=20% of basic DA=30% of basic LOAN=Rs. 600 PF=1000

AIM:

To create an Employee class that calculates and displays an employee's net salary based on their job category, with salary components and deductions, and demonstrates encapsulation and methods.

PROGRAM:
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Assignment1_Q8
{

    class Employee
    {
        // Data members for employee details and salary components
        private int empno;
        private string empname;
```

```csharp
    private int job_catg;
    private decimal basic;
    private decimal hra;
    private decimal da;
    private decimal pf;
    private decimal loan;
    private decimal netSalary;

    // Constructor to initialize salary components
    public Employee()
    {
        // Initialization of salary components based on job category will be done
in calculateSalary()
        hra = 0;
        da = 0;
        pf = 0;
        loan = 0;
        netSalary = 0;
    }

    // Method to get employee details
    public void Input()
    {
        Console.Write("Enter Employee Number: ");
        empno = int.Parse(Console.ReadLine());

        Console.Write("Enter Employee Name: ");
        empname = Console.ReadLine();

        Console.Write("Enter Job Category (1 for Table-I, 2 for Table-II): ");
        job_catg = int.Parse(Console.ReadLine());
    }

    // Method to calculate salary based on job category
    public void CalculateSalary()
    {
        if (job_catg == 1)
        {
            // Table-I calculations
            basic = 8000;
```

```csharp
            hra = 0.10m * basic;
            da = 0.20m * basic;
            loan = 300;
            pf = 500;
        }
        else if (job_catg == 2)
        {
            // Table-II calculations
            basic = 15000;
            hra = 0.20m * basic;
            da = 0.30m * basic;
            loan = 600;
            pf = 1000;
        }
        else
        {
            Console.WriteLine("Invalid Job Category!");
            return;
        }

        // Calculate net salary
        netSalary = basic + hra + da - (pf + loan);
    }

    // Method to display employee details and salary
    public void Display()
    {
        Console.WriteLine("\nEmployee Details:");
        Console.WriteLine($"Employee Number: {empno}");
        Console.WriteLine($"Employee Name: {empname}");
        Console.WriteLine($"Job Category: {job_catg}");
        Console.WriteLine($"Basic Salary: Rs. {basic}");
        Console.WriteLine($"HRA: Rs. {hra}");
        Console.WriteLine($"DA: Rs. {da}");
        Console.WriteLine($"Loan Deduction: Rs. {loan}");
        Console.WriteLine($"PF Deduction: Rs. {pf}");
        Console.WriteLine($"Net Salary: Rs. {netSalary}");
    }
}
class Program
```

```csharp
    {
        static void Main(string[] args)
        {
            // Create an instance of the Employee class
            Employee employee = new Employee();

            // Get employee details
            employee.Input();

            // Calculate salary
            employee.CalculateSalary();

            // Display employee details and net salary
            employee.Display();

            // Wait for user input before closing
            Console.WriteLine("\nPress any key to exit...");
            Console.ReadKey();
        }
    }
}
```

INPUT:
Enter Employee Number: 21
Enter Employee Name: Surya
Enter Job Category (1 for Table-I, 2 for Table-II): 1

OUTPUT:
Employee Details:
Employee Number: 21
Employee Name: Surya
Job Category: 1
Basic Salary: Rs. 8000
HRA: Rs. 800.00
DA: Rs. 1600.00
Loan Deduction: Rs. 300
PF Deduction: Rs. 500
Net Salary: Rs. 9600.00

BY:
VENKATESHWARAN S
73772226159
III – B.TECH AI&DS