

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 create a C# program that initializes a two-dimensional array and prints all the elements in the array on the same line, separated by spaces.

PROGRAM AND OUTPUT:



```
1 using System;
2
3 class TwoDimArr {
4     static void Main() {
5         // Declare and initialize a 2D array
6         int[,] array = {
7             {1, 2, 3},
8             {4, 5, 6},
9             {7, 8, 9}
10        };
11
12        // Loop through the array and print the elements
13        for (int i = 0; i < array.GetLength(0); i++) {
14            for (int j = 0; j < array.GetLength(1); j++) {
15                Console.Write(array[i, j] + " ");
16            }
17            Console.WriteLine(); // New line after each row
18        }
19    }
20 }
```

STDIN

Input for the program (Optional)

Output:

```
1 2 3
4 5 6
7 8 9
```

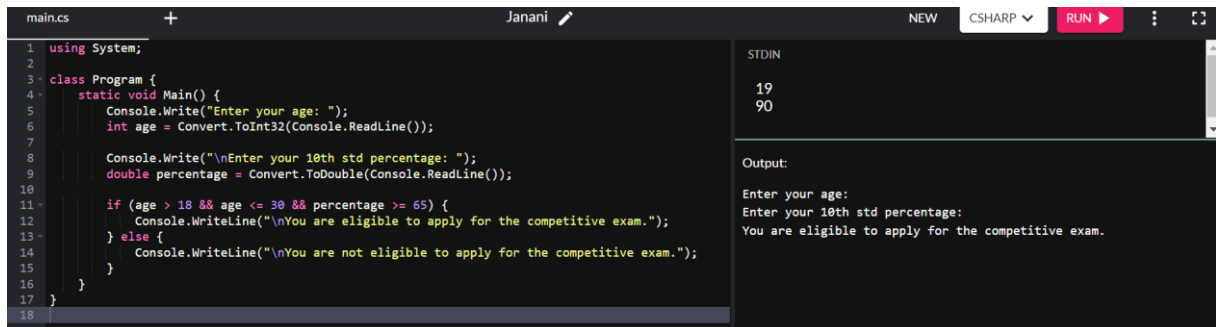
2. Aravind wants to apply for competitive exam. He needs to know whether he is eligible to apply. The eligibility criteria is given below:
 - Age should be greater than 18 years, but not more than 30.
 - The candidate should have passed 10 std with a minimum pass percentage of 65.

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 create a C# program that checks whether Aravind is eligible to apply for a competitive exam based on age and 10th standard percentage criteria.

PROGRAM AND OUTPUT:



```
1 using System;
2
3 class Program {
4     static void Main() {
5         Console.Write("Enter your age: ");
6         int age = Convert.ToInt32(Console.ReadLine());
7
8         Console.Write("\nEnter your 10th std percentage: ");
9         double percentage = Convert.ToDouble(Console.ReadLine());
10
11         if (age > 18 && age <= 30 && percentage >= 65) {
12             Console.WriteLine("\nYou are eligible to apply for the competitive exam.");
13         } else {
14             Console.WriteLine("\nYou are not eligible to apply for the competitive exam.");
15         }
16     }
17 }
18
```

STDIN

19
90

Output:

Enter your age:
Enter your 10th std percentage:
You are eligible to apply for the competitive exam.

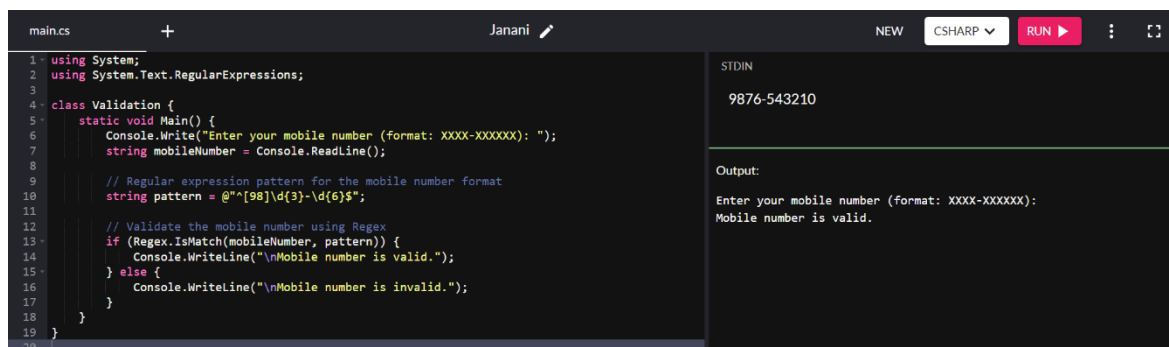
3. Design the C# console application named validation to get mobile number as input from the user. Validate the mobile number with the following cases:

- 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 create a C# console application that validates a mobile number based on specific criteria, including format, numeric content, length, and starting digit.

PROGRAM AND OUTPUT:



```
1 using System;
2 using System.Text.RegularExpressions;
3
4 class Validation {
5     static void Main() {
6         Console.Write("Enter your mobile number (format: XXXX-XXXXXX): ");
7         string mobileNumber = Console.ReadLine();
8
9         // Regular expression pattern for the mobile number format
10        string pattern = @"^[98]\d{3}-\d{6}$";
11
12        // Validate the mobile number using Regex
13        if (Regex.IsMatch(mobileNumber, pattern)) {
14            Console.WriteLine("\nMobile number is valid.");
15        } else {
16            Console.WriteLine("\nMobile number is invalid.");
17        }
18    }
19 }
20
```

STDIN

9876-543210

Output:

Enter your mobile number (format: XXXX-XXXXXX):
Mobile number is valid.

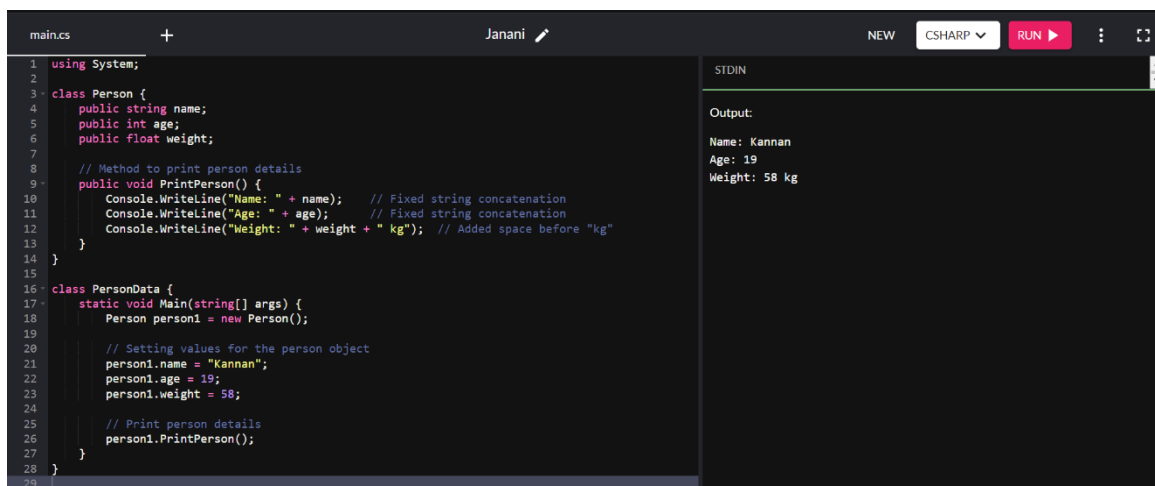
4. Write the missing code snippets and the statements in the C# program given below.

```
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 C# program that defines a Person class with attributes name, age, and weight, and a method to print these values.

PROGRAM AND OUTPUT:



The screenshot shows a C# program in Visual Studio. The code defines a `Person` class with attributes `name`, `age`, and `weight`, and a method `PrintPerson`. It also defines a `PersonData` class with a `Main` method that creates a `Person` object, sets its attributes, and calls `PrintPerson`. The output of the program is displayed in the console window.

```
1 using System;  
2  
3 class Person {  
4     public string name;  
5     public int age;  
6     public float weight;  
7  
8     // Method to print person details  
9     public void PrintPerson() {  
10        Console.WriteLine("Name: " + name); // Fixed string concatenation  
11        Console.WriteLine("Age: " + age); // Fixed string concatenation  
12        Console.WriteLine("Weight: " + weight + " kg"); // Added space before "kg"  
13    }  
14 }  
15  
16 class PersonData {  
17     static void Main(string[] args) {  
18         Person person1 = new Person();  
19  
20         // Setting values for the person object  
21         person1.name = "Kannan";  
22         person1.age = 19;  
23         person1.weight = 58;  
24  
25         // Print person details  
26         person1.PrintPerson();  
27     }  
28 }  
29
```

Output:

```
Name: Kannan  
Age: 19  
Weight: 58 kg
```

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

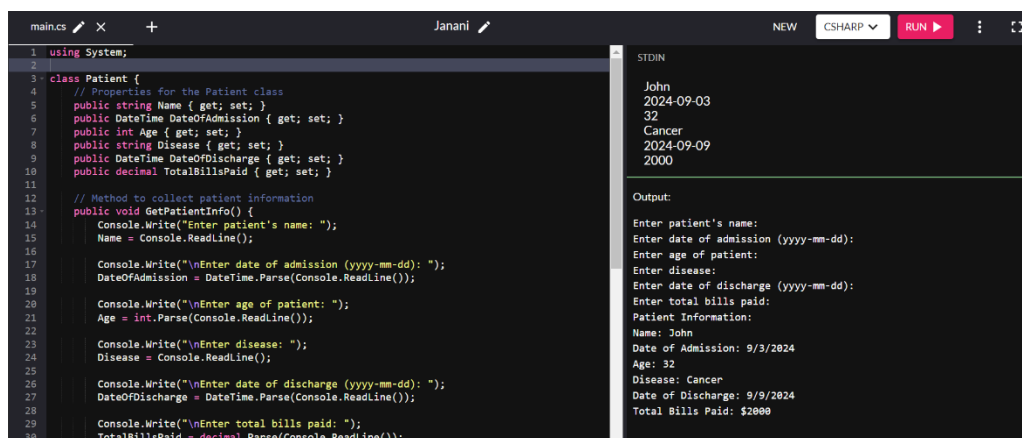
- Name of the patient
- Date of admission
- Age of patient
- Disease
- Date of discharge
- Total bills paid

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 C# console application that maintains and displays inpatient details including the patient's name, admission and discharge dates, age, disease, and total bills paid.

PROGRAM AND OUTPUT:



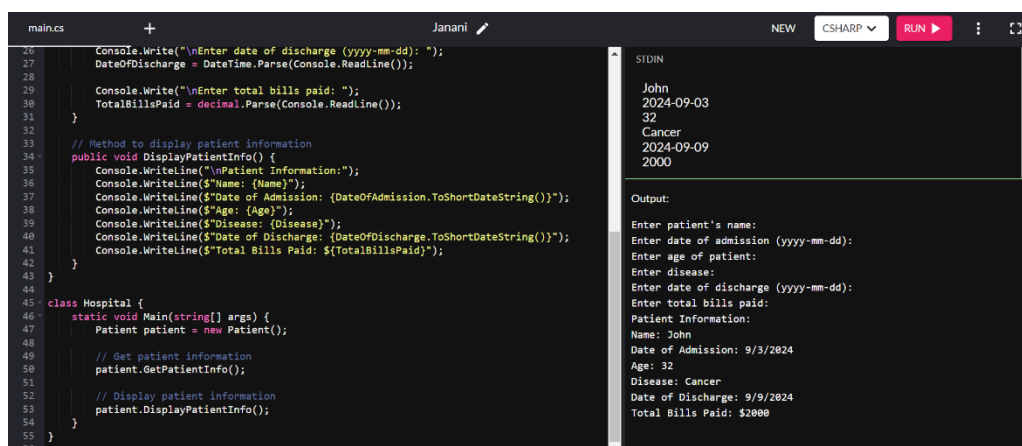
```
1 using System;
2
3 class Patient {
4     // Properties for the Patient class
5     public string Name { get; set; }
6     public DateTime DateOfAdmission { get; set; }
7     public int Age { get; set; }
8     public string Disease { get; set; }
9     public DateTime DateOfDischarge { get; set; }
10    public decimal TotalBillsPaid { get; set; }
11
12    // Method to collect patient information
13    public void GetPatientInfo() {
14        Console.Write("Enter patient's name: ");
15        Name = Console.ReadLine();
16
17        Console.Write("\nEnter date of admission (yyyy-mm-dd): ");
18        DateOfAdmission = DateTime.Parse(Console.ReadLine());
19
20        Console.Write("\nEnter age of patient: ");
21        Age = int.Parse(Console.ReadLine());
22
23        Console.Write("\nEnter disease: ");
24        Disease = Console.ReadLine();
25
26        Console.Write("\nEnter date of discharge (yyyy-mm-dd): ");
27        DateOfDischarge = DateTime.Parse(Console.ReadLine());
28
29        Console.Write("\nEnter total bills paid: ");
30        TotalBillsPaid = decimal.Parse(Console.ReadLine());
31    }
32}
```

STDIN

```
John
2024-09-03
32
Cancer
2024-09-09
2000
```

Output:

```
Enter patient's name:
Enter date of admission (yyyy-mm-dd):
Enter age of patient:
Enter disease:
Enter date of discharge (yyyy-mm-dd):
Enter total bills paid:
Patient Information:
Name: John
Date of Admission: 9/3/2024
Age: 32
Disease: Cancer
Date of Discharge: 9/9/2024
Total Bills Paid: $2000
```



```
26 Console.Write("\nEnter date of discharge (yyyy-mm-dd): ");
27 DateOfDischarge = DateTime.Parse(Console.ReadLine());
28
29 Console.Write("\nEnter total bills paid: ");
30 TotalBillsPaid = decimal.Parse(Console.ReadLine());
31 }
32
33 // Method to display patient information
34 public void DisplayPatientInfo() {
35     Console.WriteLine("\nPatient Information:");
36     Console.WriteLine($"Name: {Name}");
37     Console.WriteLine($"Date of Admission: {DateOfAdmission.ToShortDateString()}");
38     Console.WriteLine($"Age: {Age}");
39     Console.WriteLine($"Disease: {Disease}");
40     Console.WriteLine($"Date of Discharge: {DateOfDischarge.ToShortDateString()}");
41     Console.WriteLine($"Total Bills Paid: ${TotalBillsPaid}");
42 }
43
44 class Hospital {
45     static void Main(string[] args) {
46         Patient patient = new Patient();
47
48         // Get patient information
49         patient.GetPatientInfo();
50
51         // Display patient information
52         patient.DisplayPatientInfo();
53     }
54 }
55
56 }
```

STDIN

```
John
2024-09-03
32
Cancer
2024-09-09
2000
```

Output:

```
Enter patient's name:
Enter date of admission (yyyy-mm-dd):
Enter age of patient:
Enter disease:
Enter date of discharge (yyyy-mm-dd):
Enter total bills paid:
Patient Information:
Name: John
Date of Admission: 9/3/2024
Age: 32
Disease: Cancer
Date of Discharge: 9/9/2024
Total Bills Paid: $2000
```

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 C# program that uses operator overloading to add two vectors and print the resulting vector.

PROGRAM AND OUTPUT:

The image displays two screenshots of a C# IDE, likely Visual Studio, showing the implementation of a vector class and its addition using operator overloading.

Top Screenshot: The code defines a `Vector` class with `X` and `Y` properties. It implements the `+` operator to add two vectors. The `Display` method prints the vector components. The `Main` method prompts the user to enter the x component of the first vector.

```
1 using System;
2
3 class Vector
4 {
5     public int X { get; set; }
6     public int Y { get; set; }
7
8     public Vector(int x, int y)
9     {
10         X = x;
11         Y = y;
12     }
13
14     public static Vector operator +(Vector v1, Vector v2)
15     {
16         return new Vector(v1.X + v2.X, v1.Y + v2.Y);
17     }
18
19     public void Display()
20     {
21         Console.WriteLine($"({X}, {Y})");
22     }
23 }
24 class Program
25 {
26     static void Main(string[] args)
27     {
28         Console.Write("\nEnter the x component of the first vector: ");
29         int x1 = int.Parse(Console.ReadLine());
30     }
31 }
```

Bottom Screenshot: This screenshot shows the continuation of the `Main` method, where the y component of the first vector and both components of the second vector are entered. It then creates two `Vector` objects, adds them, and displays the result.

```
19 public void Display()
20 {
21     Console.WriteLine($"({X}, {Y})");
22 }
23 }
24 class Program
25 {
26     static void Main(string[] args)
27     {
28         Console.Write("\nEnter the x component of the first vector: ");
29         int x1 = int.Parse(Console.ReadLine());
30
31         Console.Write("\nEnter the y component of the first vector: ");
32         int y1 = int.Parse(Console.ReadLine());
33
34         Console.Write("\nEnter the x component of the second vector: ");
35         int x2 = int.Parse(Console.ReadLine());
36
37         Console.Write("\nEnter the y component of the second vector: ");
38         int y2 = int.Parse(Console.ReadLine());
39
40         Vector vector1 = new Vector(x1, y1);
41         Vector vector2 = new Vector(x2, y2);
42
43         Vector result = vector1 + vector2;
44
45         Console.Write("\nSum of the vectors: ");
46         result.Display();
47     }
48 }
```

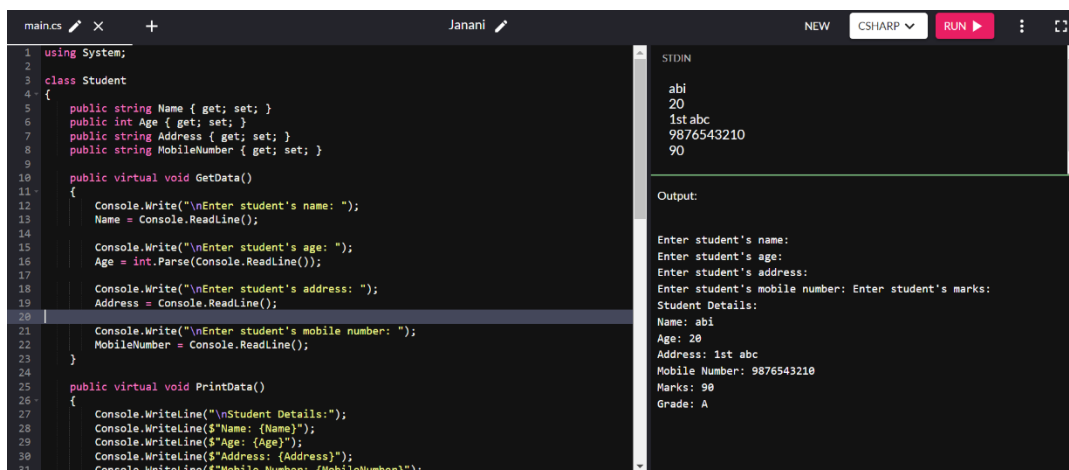
Output: The output shows the user input for the first vector (2, 3) and the second vector (4, 5), followed by the sum of the vectors: (6, 8).

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 getDate() 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 C# program that manages and displays student details including basic information and marks, with functionality to compute and display the student's grade. The program uses class inheritance and method overriding.

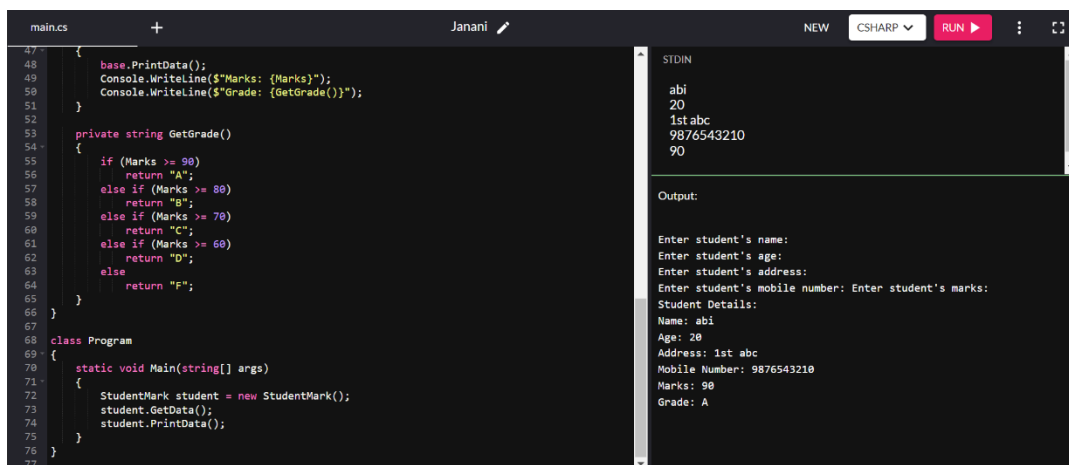
PROGRAM AND OUTPUT:



```
1 using System;
2
3 class Student
4 {
5     public string Name { get; set; }
6     public int Age { get; set; }
7     public string Address { get; set; }
8     public string MobileNumber { get; set; }
9
10    public virtual void GetData()
11    {
12        Console.WriteLine("Enter student's name: ");
13        Name = Console.ReadLine();
14
15        Console.WriteLine("Enter student's age: ");
16        Age = int.Parse(Console.ReadLine());
17
18        Console.WriteLine("Enter student's address: ");
19        Address = Console.ReadLine();
20
21        Console.WriteLine("Enter student's mobile number: ");
22        MobileNumber = Console.ReadLine();
23    }
24
25    public virtual void PrintData()
26    {
27        Console.WriteLine("\nStudent Details:");
28        Console.WriteLine($"Name: {Name}");
29        Console.WriteLine($"Age: {Age}");
30        Console.WriteLine($"Address: {Address}");
31        Console.WriteLine($"Mobile Number: {MobileNumber}");
32    }
33 }
```

Output:

```
Enter student's name:
Enter student's age:
Enter student's address:
Enter student's mobile number: Enter student's marks:
Student Details:
Name: abi
Age: 20
Address: 1st abc
Mobile Number: 9876543210
Marks: 90
Grade: A
```



```
47 {
48     base.PrintData();
49     Console.WriteLine($"Marks: {Marks}");
50     Console.WriteLine($"Grade: {GetGrade()}");
51 }
52
53 private string GetGrade()
54 {
55     if (Marks >= 90)
56         return "A";
57     else if (Marks >= 80)
58         return "B";
59     else if (Marks >= 70)
60         return "C";
61     else if (Marks >= 60)
62         return "D";
63     else
64         return "F";
65 }
66 }
67
68 class Program
69 {
70     static void Main(string[] args)
71     {
72         StudentMark student = new StudentMark();
73         student.GetData();
74         student.PrintData();
75     }
76 }
77 }
```

Output:

```
Enter student's name:
Enter student's age:
Enter student's address:
Enter student's mobile number: Enter student's marks:
Student Details:
Name: abi
Age: 20
Address: 1st abc
Mobile Number: 9876543210
Marks: 90
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-I 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 a C# program that calculates and displays the net salary of an employee based on their job category using predefined salary tables. The program uses constructors for initialization and methods for input, salary calculation, and displaying details.

PROGRAM AND OUTPUT:

```
1 using System;
2
3 class Employee
4 {
5     private int jobCatg;
6     private int empNo;
7     private string empName;
8     private decimal basicSalary;
9     private decimal hra;
10    private decimal da;
11    private decimal pf;
12    private decimal loan;
13    private decimal netSalary;
14
15    public Employee(decimal basicSalary, decimal hra, decimal da, decimal pf, decimal loan)
16    {
17        this.basicSalary = basicSalary;
18        this.hra = hra;
19        this.da = da;
20        this.pf = pf;
21        this.loan = loan;
22    }
23
24    public void Input()
25    {
26        Console.Write("Enter employee number: ");
27        empNo = int.Parse(Console.ReadLine());
28
29        Console.Write("Enter employee name: ");
30        empName = Console.ReadLine();
31    }
```

STDIN

```
7
emp1
2
```

Output:

```
Enter employee number: Enter employee name: Enter job category (1 f
Employee Details:
Employee Number: 7
Employee Name: emp1
Basic Salary: Rs. 15000
HRA: Rs. 3000.00
DA: Rs. 4500.00
Loan: Rs. 600
PF: Rs. 1000
Net Salary: Rs. 20900.00
```

```
main.cs + Janani NEW CSHARP RUN
32 Console.WriteLine("Enter job category (1 for Table-I, 2 for Table-II): ");
33 jobCatg = int.Parse(Console.ReadLine());
34 }
35
36 public void CalculateSalary()
37 {
38     switch (jobCatg)
39     {
40         case 1:
41             basicSalary = 8000;
42             hra = 0.10m * basicSalary;
43             da = 0.20m * basicSalary;
44             loan = 300;
45             pf = 500;
46             break;
47
48         case 2:
49             basicSalary = 15000;
50             hra = 0.20m * basicSalary;
51             da = 0.30m * basicSalary;
52             loan = 600;
53             pf = 1000;
54             break;
55
56         default:
57             Console.WriteLine("Invalid job category.");
58             return;
59     }
60
61     netSalary = basicSalary + hra + da - loan - pf;

```

STDIN

```
7
emp1
2

```

Output:

```
Enter employee number: Enter employee name: Enter job category (1 f
Employee Details:
Employee Number: 7
Employee Name: emp1
Basic Salary: Rs. 15000
HRA: Rs. 3000.00
DA: Rs. 4500.00
Loan: Rs. 600
PF: Rs. 1000
Net Salary: Rs. 20900.00

```

```
main.cs + Janani NEW CSHARP RUN
32 Console.WriteLine("Enter job category (1 for Table-I, 2 for Table-II): ");
33 jobCatg = int.Parse(Console.ReadLine());
34 }
35
36 public void CalculateSalary()
37 {
38     switch (jobCatg)
39     {
40         case 1:
41             basicSalary = 8000;
42             hra = 0.10m * basicSalary;
43             da = 0.20m * basicSalary;
44             loan = 300;
45             pf = 500;
46             break;
47
48         case 2:
49             basicSalary = 15000;
50             hra = 0.20m * basicSalary;
51             da = 0.30m * basicSalary;
52             loan = 600;
53             pf = 1000;
54             break;
55
56         default:
57             Console.WriteLine("Invalid job category.");
58             return;
59     }
60
61     netSalary = basicSalary + hra + da - loan - pf;

```

STDIN

```
7
emp1
2

```

Output:

```
Enter employee number: Enter employee name: Enter job category (1 f
Employee Details:
Employee Number: 7
Employee Name: emp1
Basic Salary: Rs. 15000
HRA: Rs. 3000.00
DA: Rs. 4500.00
Loan: Rs. 600
PF: Rs. 1000
Net Salary: Rs. 20900.00

```

BY:

JANANI S

73772226118

III – B.TECH AI&DS