C# and .NET Frameworks

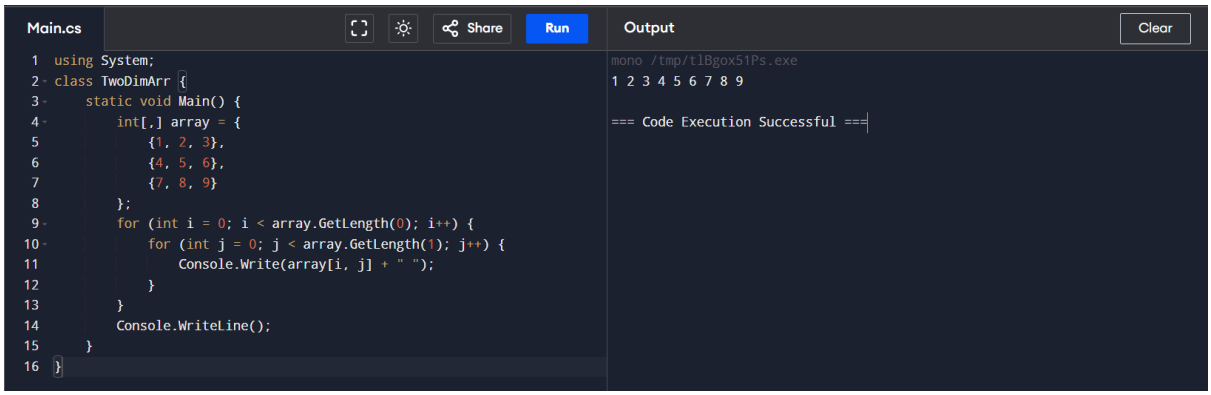
Assignment 1

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:



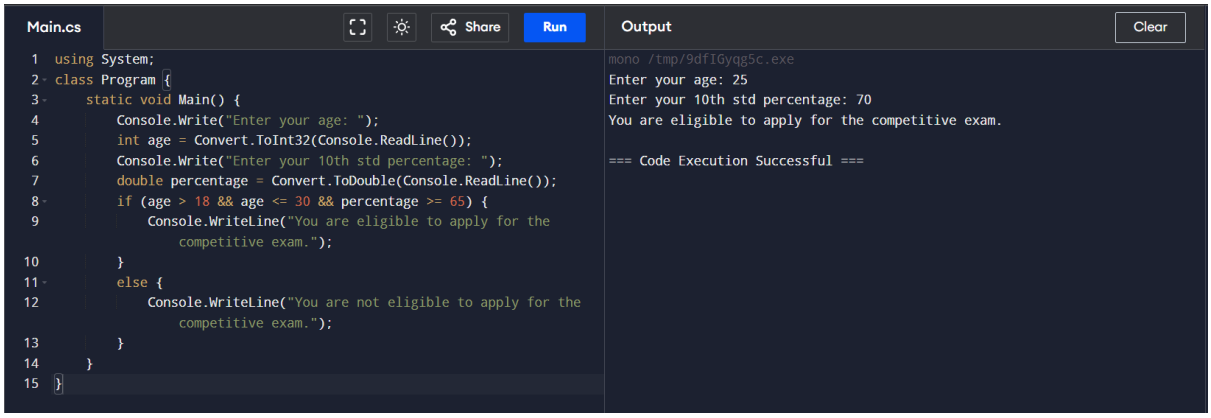
2. **Aravind wants to apply for competitive exam. He needs to know whether he is eligible to apply. The eligibility criteria is given below:**
   - **Age should be greater than 18 years, but not more than 30.**
   - **The candidate should have passed 10 std with a minimum pass percentage of 65.**

   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:

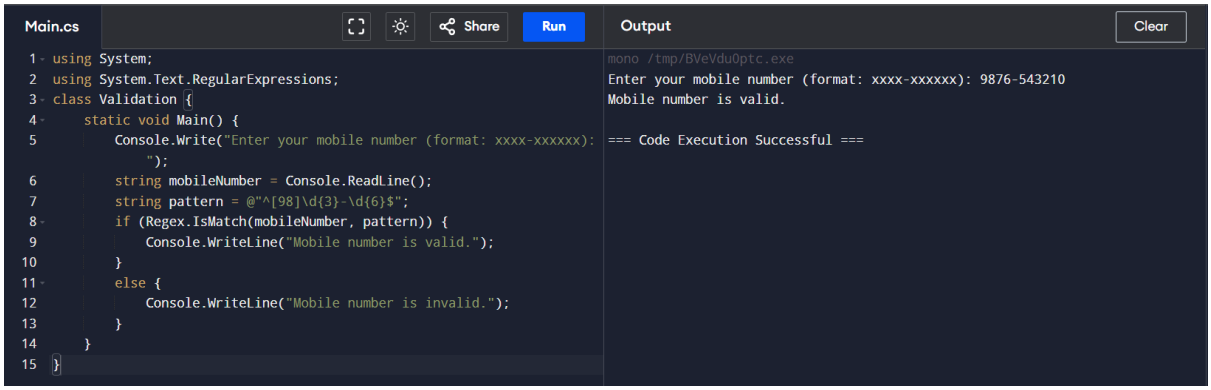


3. **Design the C# console application named validation to get mobile number as input from the user. Validate the mobile number with the following cases:**
    - **The first four number must be followed by then followed by next six numbers(eg:9894-256874)**
    - **Should contains only numbers**
    - **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:



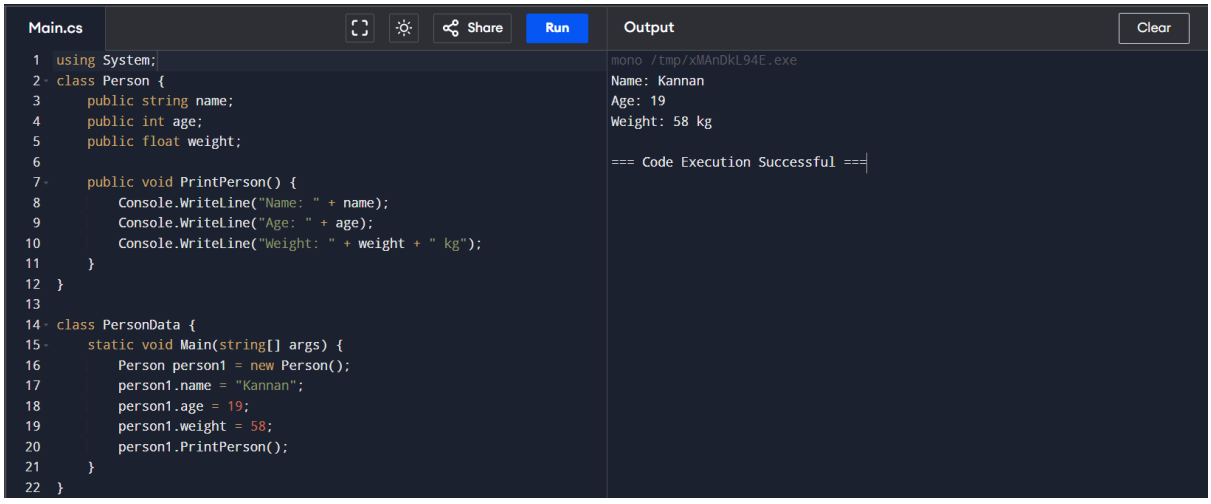
4. **Write the missing code snippets and the statements in the C# program given below.**

```
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:



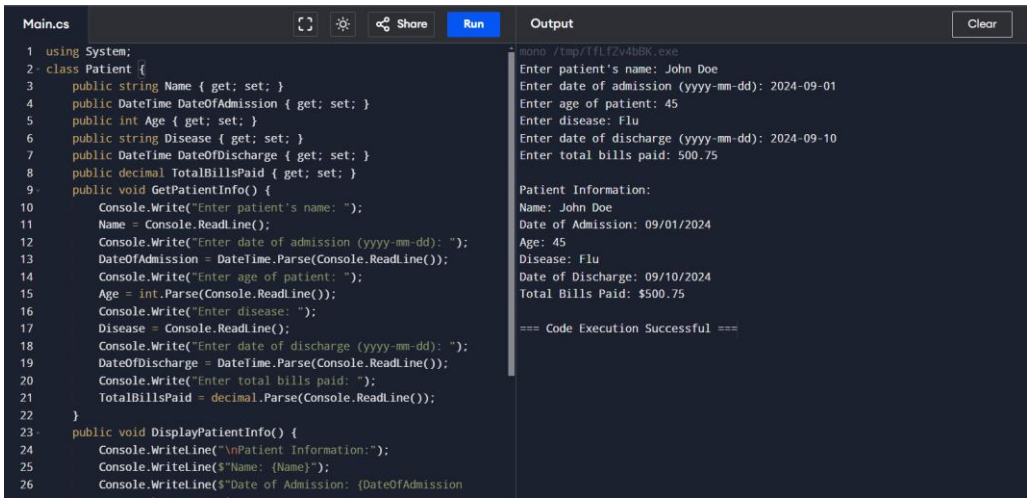
5. **A hospital wants to create a console application to maintain its impatient details. The information to store includes:**
    - **Name of the patient**
    - **Date of admission**
    - **Age of patient**
    - **Disease**
    - **Date of discharge**
    - **Total bills paid**

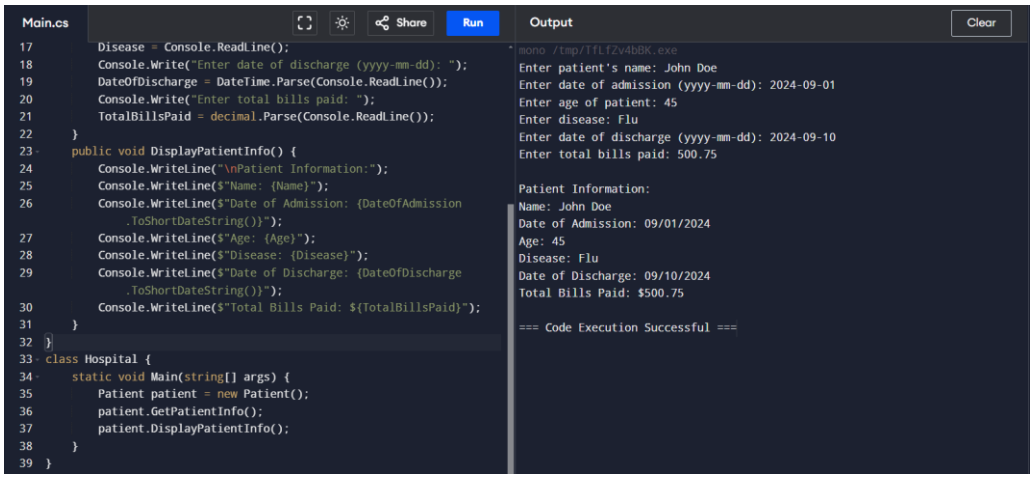
    Design the C# program with the class name patient with necessary data members to store the above information. The class should have two member functions, one to get the patients information and other to display the information. Create a main class called hospital to create necessary instances, methods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

AIM:

To create a 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:





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:

Main.cs | Share | Run

```
using System;
class Vector {
    public int X { get; set; }
    public int Y { get; set; }
    public Vector(int x, int y) {
        X = x;
        Y = y;
    }
    public static Vector operator +(Vector v1, Vector v2) {
        return new Vector(v1.X + v2.X, v1.Y + v2.Y);
    }
    public void Display() {
        Console.WriteLine($"({X}, {Y})");
    }
}
class Program {
    static void Main(string[] args) {
        Console.Write("Enter the x component of the first vector: "
            );
        int x1 = int.Parse(Console.ReadLine());
        Console.Write("Enter the y component of the first vector: "
            );
        int y1 = int.Parse(Console.ReadLine());
        Console.Write("Enter the x component of the second vector: "
            );
        int x2 = int.Parse(Console.ReadLine());
```

Output | Clear

```
mono /tmp/AnicXWXVdS.exe
Enter the x component of the first vector: 2
Enter the y component of the first vector: 3
Enter the x component of the second vector: 4
Enter the y component of the second vector: 5
Sum of the vectors: (6, 8)

=== Code Execution Successful ===
```

Main.cs | Share | Run

```
    public void Display() {
        Console.WriteLine($"({X}, {Y})");
    }
}
class Program {
    static void Main(string[] args) {
        Console.Write("Enter the x component of the first vector: "
            );
        int x1 = int.Parse(Console.ReadLine());
        Console.Write("Enter the y component of the first vector: "
            );
        int y1 = int.Parse(Console.ReadLine());
        Console.Write("Enter the x component of the second vector: "
            );
        int x2 = int.Parse(Console.ReadLine());
        Console.Write("Enter the y component of the second vector: "
            );
        int y2 = int.Parse(Console.ReadLine());
        Vector vector1 = new Vector(x1, y1);
        Vector vector2 = new Vector(x2, y2);
        Vector result = vector1 + vector2;
        Console.Write("Sum of the vectors: ");
        result.Display();
    }
}
```

Output | Clear

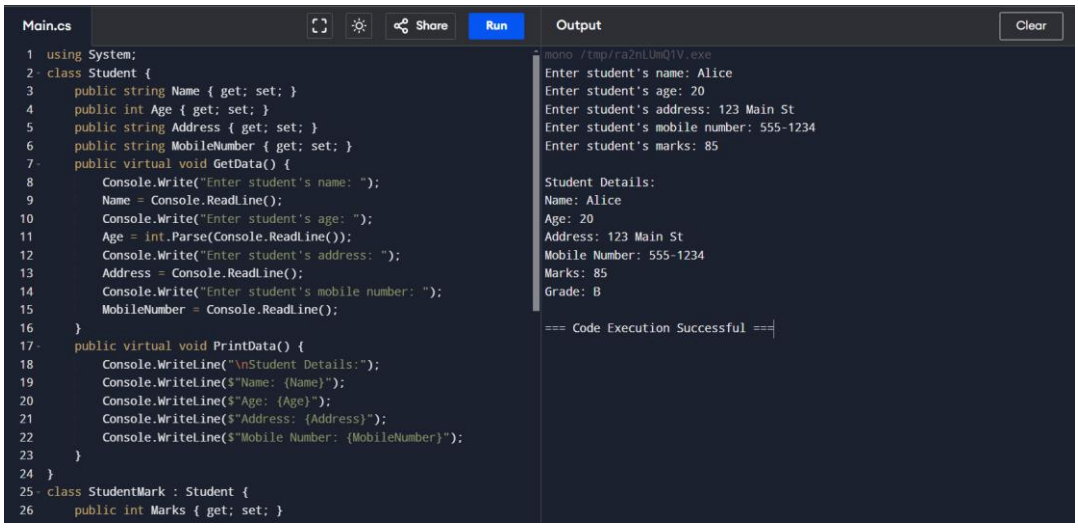
```
mono /tmp/AnicXWXVdS.exe
Enter the x component of the first vector: 2
Enter the y component of the first vector: 3
Enter the x component of the second vector: 4
Enter the y component of the second vector: 5
Sum of the vectors: (6, 8)

=== Code Execution Successful ===
```

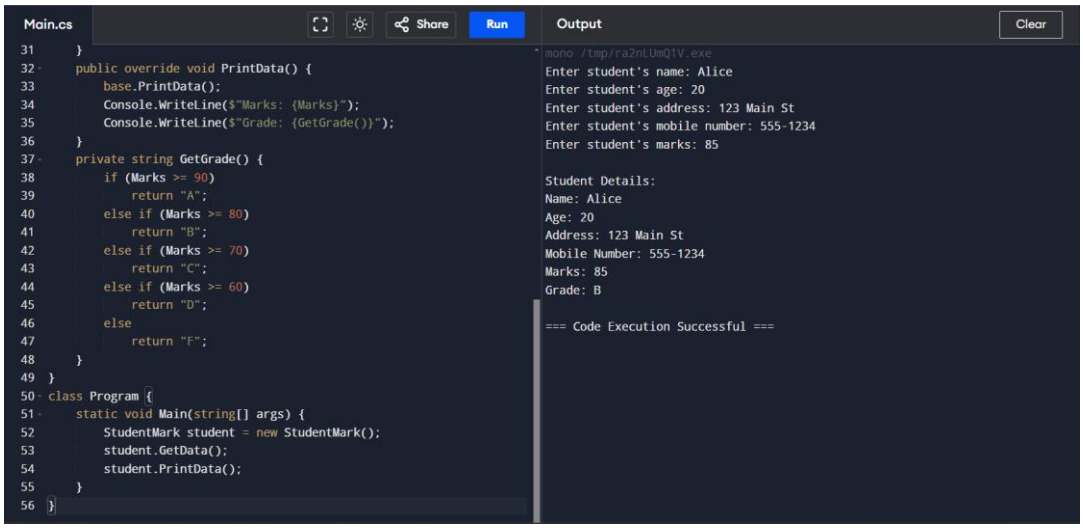
7. **Create the class student with necessary members to maintain the basic details of a student such as name, age, address and mobile number. Add method getDate() to read the basic details and printData() to print the details of the student. Inherit the student class into the sub class called studentmark 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 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:





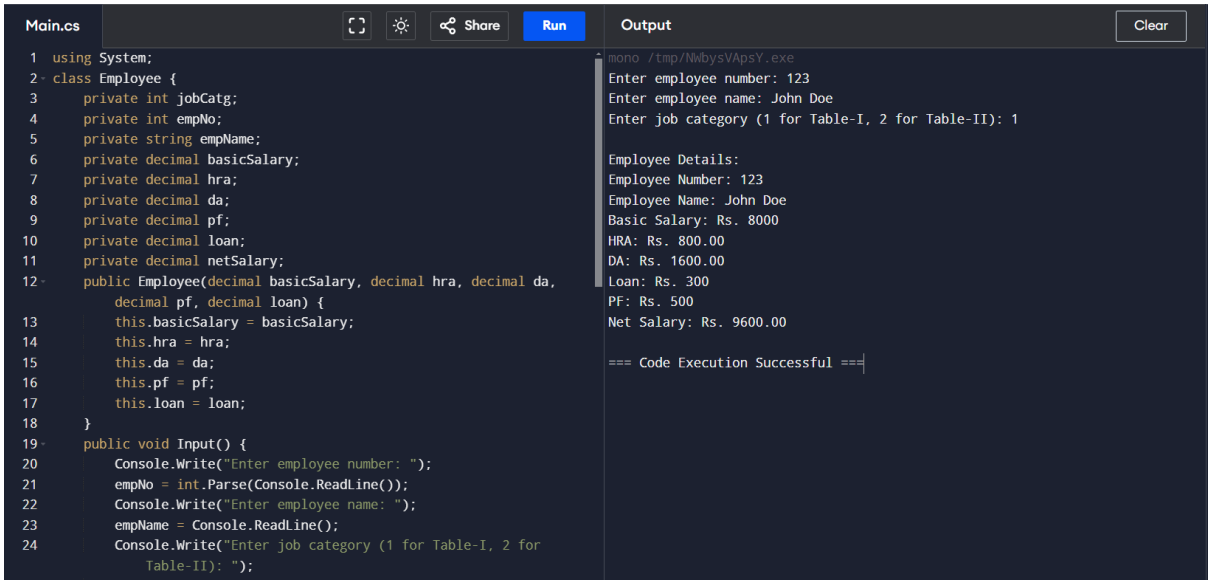
8. **Design sample C# program with class name employee to compute netsalary of the employee using the basic salary, if for the job_catg is 1 use table-l else use table-II. Use constructor to initialize basic salary,hra,da,pf and loan. The employee class should contain input() method to get input for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



Main.cs

Share Run

```
        jobCatg = int.Parse(Console.ReadLine());
    }
    public void CalculateSalary() {
        switch (jobCatg) {
            case 1:
                basicSalary = 8000;
                hra = 0.10m * basicSalary;
                da = 0.20m * basicSalary;
                loan = 300;
                pf = 500;
                break;
            case 2:
                basicSalary = 15000;
                hra = 0.20m * basicSalary;
                da = 0.30m * basicSalary;
                loan = 600;
                pf = 1000;
                break;
            default:
                Console.WriteLine("Invalid job category.");
                return;
        }
        netSalary = basicSalary + hra + da - loan - pf;
    }
    public void Display() {
        Console.WriteLine("\nEmployee Details:");
```

Output

Clear

```
mono /tmp/NWbysVApsY.exe
Enter employee number: 123
Enter employee name: John Doe
Enter job category (1 for Table-I, 2 for Table-II): 1

Employee Details:
Employee Number: 123
Employee Name: John Doe
Basic Salary: Rs. 8000
HRA: Rs. 800.00
DA: Rs. 1600.00
Loan: Rs. 300
PF: Rs. 500
Net Salary: Rs. 9600.00

=== Code Execution Successful ===
```

Main.cs

Share Run

```
            default:
                Console.WriteLine("Invalid job category.");
                return;
        }
        netSalary = basicSalary + hra + da - loan - pf;
    }
    public void Display() {
        Console.WriteLine("\nEmployee Details:");
        Console.WriteLine($"Employee Number: {empNo}");
        Console.WriteLine($"Employee Name: {empName}");
        Console.WriteLine($"Basic Salary: Rs. {basicSalary}");
        Console.WriteLine($"HRA: Rs. {hra}");
        Console.WriteLine($"DA: Rs. {da}");
        Console.WriteLine($"Loan: Rs. {loan}");
        Console.WriteLine($"PF: Rs. {pf}");
        Console.WriteLine($"Net Salary: Rs. {netSalary}");
    }
}
class Program {
    static void Main(string[] args) {
        Employee employee = new Employee(0, 0, 0, 0, 0);
        employee.Input();
        employee.CalculateSalary();
        employee.Display();
    }
}
```

Output

Clear

```
mono /tmp/NWbysVApsY.exe
Enter employee number: 123
Enter employee name: John Doe
Enter job category (1 for Table-I, 2 for Table-II): 1

Employee Details:
Employee Number: 123
Employee Name: John Doe
Basic Salary: Rs. 8000
HRA: Rs. 800.00
DA: Rs. 1600.00
Loan: Rs. 300
PF: Rs. 500
Net Salary: Rs. 9600.00

=== Code Execution Successful ===
```

BY:

VARUNA SREE N

73772226157

III – B.TECH AI&DS