C# ASSIGNMENT 01 Abivarman G

CSE 'A'

01. 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:

Write a code to develop the C# program to initialize two-dimensional array and print all the elements of the array on the same line separated with space.

PROGRAM:

```
using System;
class Program
{
   static void Main()
   {
      int[,] array = {
         { 1,2,3},
         { 4,5,6 },
         { 7,8,9 }
      };
foreach (var item in array)
{
       Console.Write(item + " ");
 }
}
```

OUTPUT:

```
1 2 3 4 5 6 7 8 9
=== Code Execution Successful ===
```

02. 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:

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.

PROGRAM:

```
using System;

class Program

{

  static void Main()

  {

    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());

    CheckEligibility(age, passPercentage);

  }

 static void CheckEligibility(int age, double passPercentage)

  {

    if (age > 18 && age <= 30 && passPercentage >= 65)

    {

      Console.WriteLine("eligible”);

    }

    else

    {

      Console.WriteLine("Not eligible");
```

```
    }
  }
```

OUTPUT:

```
Enter Aravind's age: 19
Enter Aravind's 10th standard pass percentage: 28
Not eligible

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

---

03. 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

AIM:

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

PROGRAM:

```
using System;

class Validation

{

  static void Main()

  {

    Console.Write("Enter the mobile number: ");

    string mobileNumber = Console.ReadLine();

    if (IsValidMobileNumber(mobileNumber))

    {

      Console.WriteLine("The mobile number is valid.");

    }
```

```
        else
        {
            Console.WriteLine("The mobile number is invalid.");
        }
    }
static bool IsValidMobileNumber(string mobileNumber)
    {
       if (mobileNumber.Length == 11 && mobileNumber[4] == '-' &&
            (mobileNumber[0] == '9' || mobileNumber[0] == '8'))
        {
          for (int i = 0; i < mobileNumber.Length; i++)
            {
                if (i != 4 && !char.IsDigit(mobileNumber[i]))
                {
                    return false;
                }
            }
            return true;
        }
        return false;
    }
}
```

OUTPUT:

```
Enter the mobile number: 6291-731834
The mobile number is invalid.
```

04. 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:

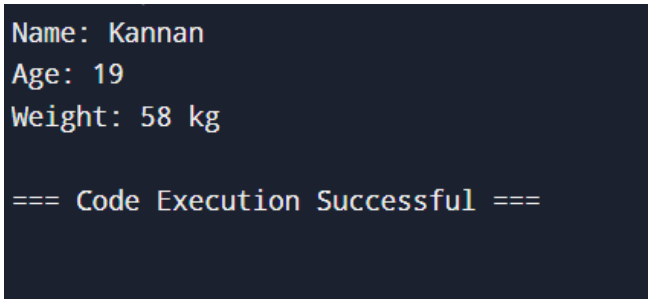
Write the missing code snippets and the statements in the C#

PROGRAM:

```
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 + " kg");

   }
```

```
}


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

OUTPUT:



05. 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:

Write a code to design the C# program with the class name patient with necessary data members to store the above information.

PROGRAM:

```
using System;

class Patient

{

   private string name;

   private string dateOfAdmission;

   private int age;

   private string disease;

   private string dateOfDischarge;

   private decimal totalBillsPaid;


   public void GetPatientInfo()

   {

      Console.Write("Enter Patient Name: ");

      name = Console.ReadLine();


      Console.Write("Enter Date of Admission: ");

      dateOfAdmission = 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: ");
        dateOfDischarge = Console.ReadLine();

        Console.Write("Enter Total Bills Paid: ");
        totalBillsPaid = decimal.Parse(Console.ReadLine());
    }

    public void DisplayPatientInfo()
    {
        Console.WriteLine("\nPatient Details:");
        Console.WriteLine($"Name: {name}");
        Console.WriteLine($"Date of Admission: {dateOfAdmission}");
        Console.WriteLine($"Age: {age}");
        Console.WriteLine($"Disease: {disease}");
        Console.WriteLine($"Date of Discharge: {dateOfDischarge}");
        Console.WriteLine($"Total Bills Paid: {totalBillsPaid:C}");
    }
}

class Hospital
{
    static void Main(string[] args)
```

```
    {
        Patient patient = new Patient();
        patient.GetPatientInfo();
        patient.DisplayPatientInfo();
        Console.WriteLine("\nPress any key to exit...");
        Console.ReadKey();
    }
}
```

OUTPUT:

```
Enter Patient Name: Abi
Enter Date of Admission: 14
Enter Age of Patient: 19
Enter Disease: fever
Enter Date of Discharge: 18
Enter Total Bills Paid: 10000.0000

Patient Details:
Name: Abi
Date of Admission: 14
Age: 19
Disease: fever
Date of Discharge: 18
Total Bills Paid: ?10,000.00
```

---

06. 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:

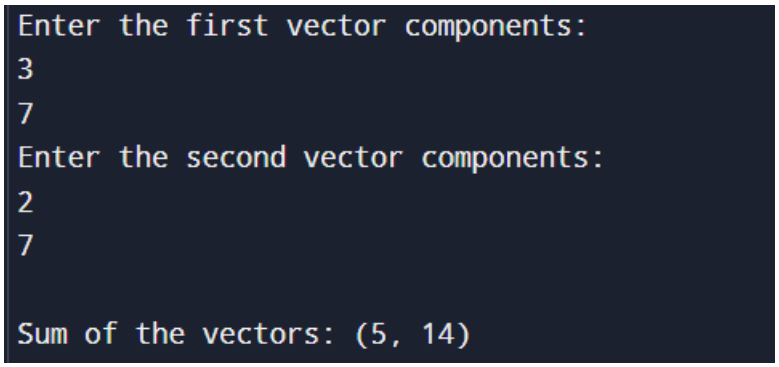
Implement the C# code to get two vector number as input, add them and print the sum as another vector.

PROGRAM:

```
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);
   }
}
class Program
{
   static void Main(string[] args)
   {
      Console.WriteLine("Enter the first vector:");
      Vector vector1 = new Vector(int.Parse(Console.ReadLine()),
int.Parse(Console.ReadLine()));
      Console.WriteLine("Enter the second vector:");
      Vector vector2 = new Vector(int.Parse(Console.ReadLine()),
int.Parse(Console.ReadLine()));
      Vector sumVector = vector1 + vector2;
      Console.WriteLine($"\nSum of the vectors: {sumVector.X} {sumVector.Y}");
      Console.ReadKey();
   }
```

}

OUTPUT:



07. 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 studentmark 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 studentmain class to access the member of both the classes.

AIM:

Write a C# code to Create the class student with necessary members to maintain the basic details of a student such as name, age, address and mobile number

PROGRAM:

```
using System;

class Student

{

   protected string name;

   protected int age;

   protected string address;

   protected string mobileNumber;

   public virtual void GetData()

   {
```

```
        Console.Write("Enter 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();
    }
public virtual void PrintData()
    {
        Console.WriteLine($"Name: {name}, Age: {age}, Address: {address}, Mobile:
{mobileNumber}");
    }
}
class StudentMark : Student
{
    private int marks;
    public override void GetData()
    {
        base.GetData();
        Console.Write("Enter Marks: ");
        marks = int.Parse(Console.ReadLine());
    }
    public override void PrintData()
    {
        base.PrintData();
        Console.WriteLine($"Marks: {marks}, Grade: {FindGrade()}");
    }
    private string FindGrade()
```

```
    {
        if (marks >= 90) return "A";
        if (marks >= 75) return "B";
        if (marks >= 60) return "C";
        if (marks >= 50) return "D";
        return "F";
    }
}
class Program
{
    static void Main(string[] args)
    {
        StudentMark student = new StudentMark();
        student.GetData();
        student.PrintData();
        Console.ReadKey();
    }
}
```

OUTPUT:

```
Enter Name: Abivarman
Enter Age: 20
Enter Address: 125 west
Enter Mobile Number: 7321035659
Enter Marks: 89
Name: Abivarman, Age: 20, Address: 125 west, Mobile: 7321035659
Marks: 89, Grade: B
```

08. 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:

Design sample C# program with class name employee to compute netsalary of the employee using the basic salary.

PROGRAM:

```
using System;
{
   class Employee
   {
      private int empno;
      private string empname;
      private int job_catg;
      private decimal basic;
      private decimal hra;
      private decimal da;
      private decimal pf;
      private decimal loan;
      private decimal netSalary;
```

```
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());
}

public void CalculateSalary()
{
    if (job_catg == 1)
    {
        basic = 8000;
        hra = 0.10m * basic;
        da = 0.20m * basic;
        loan = 300;
        pf = 500;
    }
    else if (job_catg == 2)
    {
        basic = 15000;
        hra = 0.20m * basic;
        da = 0.30m * basic;
        loan = 600;
        pf = 1000;
    }
    else
```

```
            {
                Console.WriteLine("Invalid Job Category!");
                return;
            }

            netSalary = basic + hra + da - (pf + loan);
        }
            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
    {
        static void Main(string[] args)
        {
            Employee employee = new Employee()
            employee.Input();
            employee.CalculateSalary();
            employee.Display();
            Console.WriteLine("\nPress any key to exit...");
```

```
            Console.ReadKey();
        }
    }
}
```

OUTPUT:

```
Enter Employee Number: 02
Enter Employee Name: Abivarman
Enter Job Category (1 for Table-I, 2 for Table-II): 2

Employee Details:
Employee Number: 2
Employee Name: Abivarman
Job Category: 2
Basic Salary: Rs. 15000
HRA: Rs. 3000.00
DA: Rs. 4500.00
Loan Deduction: Rs. 600
PF Deduction: Rs. 1000
Net Salary: Rs. 20900.00
```