Question -1

Design a class with private instance variables and initialize them using a constructor.

About program:

this example, the **`MyClass1`** class has **two private instance variables** (**`integerValue`** and **`stringValue`**). The constructor **`MyClass1(int intValue, string strValue)`** initializes these variables when an object of the class is created. The `DisplayValues` method is just an example of how you might access the private variables from within the class.

In the **`Main`** method of the **`Program`** class, an instance of **`MyClass1`** is created and its constructor is called with specific values. Then, the **`DisplayValues`** method is called to show the initialized values

—---

Question -2

Create a class with a constructor that calls another constructor using this() keyword.

About program:

2.1) this example, the **`MyClass2`** class has two constructors

2.2) The first constructor takes no parameters and uses the **`this()`** keyword to call the second constructor, providing default values.

2.3) The second constructor takes parameters and initializes the private variables.

2.4) When you create an instance of **`MyClass2`** using the default constructor (**`new MyClass2()`**), it will call the parameterized constructor with default values.

2.5) The **`DisplayValues`** method then displays the initialized values.

—---
-

Question 3:

—-----------------

```
using System;

public class Person
{
    public string Name { get; set; }
    public int Age { get; set; }

    // Copy constructor
    public Person(Person other)
    {
        if (other != null)
        {
            Name = other.Name;
            Age = other.Age;
        }
    }

    public void DisplayInfo()
    {
        Console.WriteLine($"Name: {Name}, Age: {Age}");
    }
}

class Program
{
    static void Main()
    {
        // Create an original Person object
        Person originalPerson = new Person
        {
            Name = "Alice",
            Age = 30
        };

        // Create a copy of the originalPerson using the copy constructor
        Person copiedPerson = new Person(originalPerson);

        // Modify the originalPerson
        originalPerson.Name = "Bob";
        originalPerson.Age = 35;

        // Display information about both objects
        Console.WriteLine("Original Person:");
        originalPerson.DisplayInfo();

        Console.WriteLine("\nCopied Person:");
        copiedPerson.DisplayInfo();
    }
}
```

Question 4:
—-----------------
Create an class Sim should have the following properties:

Instance Variables:

SimNumber:long
SimType:string

Implement two constructors for the Sim class:

1.A parameterized constructor that allows you to set the SimNumber and SimType
2.A Copy constructor copy the data to copy constructor

Methods:
Name:DisplayInfo
ReturnType:string
AccessModifier:public
return the data of the Sim

If you change the sim type for example before simtype is airtel and changed to jio update the sim type

MethodName:UpdateSimtype
Parameters:NewSimType (string)
Returntype:void
AccessModifier:public

update the sim type.

Create another class SimDetails

create object for Sim class pass the values by using constructor and also call the UpdateSimtype method.
Display the sim information using copy constructor and also display details after update
copy constructor details don't has to change it should be having old details
User has to enter the values.