Student Information System (SIS)
C# OOPS

Task 1: Define Classes Define the following classes based on the domain description:

Student class with the following attributes:

- **Student ID**
- **First Name**
- **Last Name**
- **Date of Birth**
- **Email**
- **Phone Number**

```
namespace SISApp
{
   public class Student
   {
      public int StudentId { get; set; }
      public string FirstName { get; set; }
      public string LastName { get; set; }
      public String DateOfBirth { get; set; }
      public string Email { get; set; }
      public long PhoneNumber { get; set; }
   }
}
```

Course class with the following attributes:

• Course ID

• Course Name

• Course Code

• Instructor Name

```
namespace SISApp
{
   public class Course
   {
      public int CourseId { get; set; }
      public string CourseName { get; set; }
      public string CourseCode { get; set; }
      public string InstructorName { get; set; }
   }
}
```

Enrollment class to represent the relationship between students and courses. It should have attributes:

• Enrollment ID

• Student ID (reference to a Student)

• Course ID (reference to a Course)

• Enrollment Date

```
namespace SISApp
{
   public class Enrollment
   {
      public int EnrollmentId { get; set; }
      public int StudentId { get; set; }
      public int CourseId { get; set; }
      public String EnrollmentDate { get; set; }
```

```
    }
}
```

Teacher class with the following attributes:

- **Teacher ID**
- **First Name**
- **Last Name**
- **Email**

```
namespace SISApp
{
    public class Teacher
    {
        public int TeacherId { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Email { get; set; }
    }
}
```

Payment class with the following attributes:

- **Payment ID**
- **Student ID (reference to a Student)**
- **Amount**
- **Payment Date**

```
namespace SISApp
{
    public class Payment
    {
        public int PaymentId { get; set; }
        public int StudentId { get; set; }
        public decimal Amount { get; set; }
        public String PaymentDate { get; set; }
```

```
    }
}
```

Task 2: Implement Constructors

Implement constructors for each class to initialize their attributes. Constructors are special methods that are called when an object of a class is created. They are used to set initial values for the attributes of the class. Below are detailed instructions on how to implement constructors for each class in your Student Information System (SIS) assignment:

Student Class Constructor

In the Student class, you need to create a constructor that initializes the attributes of a student when an instance of the Student class is created

SIS Class Constructor

If you have a class that represents the Student Information System itself (e.g., SIS class), you may also

implement a constructor for it. This constructor can be used to set up any initial configuration for the SIS.

Repeat the above process for each class Course, Enrollment, Teacher, Payment by defining constructors that initialize their respective attributes.

1. Student Class – Add Constructor

```
namespace SISApp
{
    public class Student
    {
        public int StudentId { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string DateOfBirth { get; set; }
        public string Email { get; set; }
        public long PhoneNumber { get; set; }
```

```
        // Constructor
        public Student(int studentId, string firstName, string lastName, String dateOfBirth,
string email, long PhoneNumber)
        {
            StudentId = studentId;
            FirstName = firstName;
            LastName = lastName;
            DateOfBirth = dateOfBirth;
            Email = email;
            PhoneNumber = phoneNumber;
        }
    }
}
```

2. Course Class – Add Constructor

```
namespace SISApp
{
    public class Course
    {
        public int CourseId { get; set; }
        public string CourseName { get; set; }
        public string CourseCode { get; set; }
        public string InstructorName { get; set; }

        // Constructor
        public Course(int courseId, string courseName, string courseCode, string
instructorName)
        {
            CourseId = courseId;
            CourseName = courseName;
            CourseCode = courseCode;
```

```
            InstructorName = instructorName;
        }
    }
}
```

3. Enrollment Class – Add Constructor

```
namespace SISApp
{
    public class Enrollment
    {
        public int EnrollmentId { get; set; }
        public int StudentId { get; set; }
        public int CourseId { get; set; }
        public string EnrollmentDate { get; set; }

        // Constructor
        public Enrollment(int enrollmentId, int studentId, int courseId, String enrollmentDate)
        {
            EnrollmentId = enrollmentId;
            StudentId = studentId;
            CourseId = courseId;
            EnrollmentDate = enrollmentDate;
        }
    }
}
```

4. Teacher Class – Add Constructor

```
namespace SISApp
{
   public class Teacher
   {
      public int TeacherId { get; set; }
      public string FirstName { get; set; }
      public string LastName { get; set; }
      public string Email { get; set; }

      // Constructor
      public Teacher(int teacherId, string firstName, string lastName, string email)
      {
         TeacherId = teacherId;
         FirstName = firstName;
         LastName = lastName;
         Email = email;
      }
   }
}
```

5. Payment Class – Add Constructor

```
namespace SISApp
{
   public class Payment
   {
      public int PaymentId { get; set; }
      public int StudentId { get; set; }
      public decimal Amount { get; set; }
      public string PaymentDate { get; set; }
```

```
        // Constructor
        public Payment(int paymentId, int studentId, decimal amount, String paymentDate)
        {
            PaymentId = paymentId;
            StudentId = studentId;
            Amount = amount;
            PaymentDate = paymentDate;
        }
    }
}
```

Task 3: Implement Methods

Student Class:

• EnrollInCourse(course: Course): Enrolls the student in a course.

• UpdateStudentInfo(firstName: string, lastName: string, dateOfBirth: String, email: string,

phoneNumber: string): Updates the student's information.

• MakePayment(amount: decimal, paymentDate: String): Records a payment made by the

student.

• DisplayStudentInfo(): Displays detailed information about the student.

• GetEnrolledCourses(): Retrieves a list of courses in which the student is enrolled.

• GetPaymentHistory(): Retrieves a list of payment records for the student.

```
namespace SISApp
{
   public class Student
   {
      public int StudentId { get; set; }
      public string FirstName { get; set; }
      public string LastName { get; set; }
      public string DateOfBirth { get; set; }
      public string Email { get; set; }
      public long PhoneNumber { get; set; }

      public Student(int studentId, string firstName, string lastName, String dateOfBirth,
string email, long PhoneNumber)
      {
         StudentId = studentId;
         FirstName = firstName;
         LastName = lastName;
```

```
        DateOfBirth = dateOfBirth;
        Email = email;
        PhoneNumber = phoneNumber;
    }

    public void EnrollInCourse(int courseId)
    {
        Console.WriteLine($"Student {FirstName} enrolled in Course ID: {courseId} ");
    }

    public void UpdateStudentInfo(string firstName, string lastName, String dateOfBirth,
string email, long PhoneNumber)
    {
        FirstName = firstName;
        LastName = lastName;
        DateOfBirth = dateOfBirth;
        Email = email;
        PhoneNumber = phoneNumber;
        Console.WriteLine("Student information updated .");
    }

    public void MakePayment(decimal amount, String paymentDate)
    {
        Console.WriteLine($"Payment of ₹{amount} made on
{paymentDate.ToShortDateString()} .");
    }

    public void DisplayStudentInfo()
    {
        Console.WriteLine("===== Student Info =====");
        Console.WriteLine($"ID: {StudentId}");
```

```
            Console.WriteLine($"Name: {FirstName} {LastName}");
            Console.WriteLine($"DOB: {DateOfBirth.ToShortDateString()}");
            Console.WriteLine($"Email: {Email}");
            Console.WriteLine($"Phone: {PhoneNumber}");
        }

        public void GetEnrolledCourses()
        {
            Console.WriteLine("Returning list of enrolled courses .");
        }

        public void GetPaymentHistory()
        {
            Console.WriteLine("Returning payment history .");
        }
    }
}
```