using System;

using System.Collections.Generic;

namespace StudentInformationSystem

{

public class Student

{

private int intStudentID;

private string strName;

private double dblGPA;

public int StudentID { get { return intStudentID; } set { intStudentID = value; } }

public string Name { get { return strName; } set { strName = value; } }

public double GPA { get { return dblGPA; } set { dblGPA = value; } }

public Student(int id, string name, double gpa)

{

intStudentID = id;

strName = name;

dblGPA = gpa;

}

public override string ToString()

{

return $"{Name} (ID: {StudentID}) - GPA: {GPA}";

}

}

public class Course

{

private string strCourseCode;

private string strCourseTitle;

private int intCreditHours;

public string CourseCode { get { return strCourseCode; } set { strCourseCode = value; } }

public string CourseTitle { get { return strCourseTitle; } set { strCourseTitle = value; } }

public int CreditHours { get { return intCreditHours; } set { intCreditHours = value; } }

public Course(string code, string title, int credits)

{

strCourseCode = code;

strCourseTitle = title;

intCreditHours = credits;

}

public override string ToString()

{

return $"{CourseCode}: {CourseTitle} ({CreditHours} Credits)";

}

}

public class Enrollment

{

private Student student;

private Course course;

private DateTime dtEnrollmentDate;

public Student EnrolledStudent { get { return student; } }

public Course EnrolledCourse { get { return course; } }

public DateTime EnrollmentDate { get { return dtEnrollmentDate; } }

public Enrollment(Student student, Course course, DateTime enrollmentDate)

{

this.student = student;

this.course = course;

dtEnrollmentDate = enrollmentDate;

}

public override string ToString()

{

return $"{student.Name} enrolled in {course.CourseTitle} on {EnrollmentDate.ToShortDateString()}";

}

}

public class StudentInfoSystem

{

private List<Student> lstStudents;

private List<Course> lstCourses;

private List<Enrollment> lstEnrollments;

public StudentInfoSystem()

{

lstStudents = new List<Student>();

lstCourses = new List<Course>();

lstEnrollments = new List<Enrollment>();

}

public void AddStudent(Student student)

{

lstStudents.Add(student);

}

public void AddCourse(Course course)

{

lstCourses.Add(course);

}

public void EnrollStudent(int studentId, string courseCode)

{

Student student = lstStudents.Find(s => s.StudentID == studentId);

Course course = lstCourses.Find(c => c.CourseCode == courseCode);

if (student == null || course == null)

{

throw new Exception("Student or Course not found");

}

Enrollment enrollment = new Enrollment(student, course, DateTime.Now);

lstEnrollments.Add(enrollment);

Console.WriteLine(enrollment);

}

public void PrintAllStudents()

{

foreach (var student in lstStudents)

{

Console.WriteLine(student);

}

}

}

public class Program

{

public static void Main(string[] args)

{

StudentInfoSystem sis = new StudentInfoSystem();

// Add students

sis.AddStudent(new Student(1, "Alice Johnson", 3.8));

sis.AddStudent(new Student(2, "Bob Smith", 3.5));

// Add courses

sis.AddCourse(new Course("CS101", "Introduction to Computer Science", 3));

sis.AddCourse(new Course("MATH201", "Calculus II", 4));

// Enroll students

sis.EnrollStudent(1, "CS101");

sis.EnrollStudent(2, "MATH201");

// Print all students

sis.PrintAllStudents();

}

}

}