ECE25100 Object-Oriented Programming
Assignment 3

****

NOTE:

1. Programs *must* compile without error. If your program does not compile, you will lose **50%** of the points. Even if your program is "essentially" correct, you will still lose 50% of the points; there is no reason to turn in a program with syntax errors.
2. To get full credit for an assignment your program must completely solve the stated problem.
3. Your program should contain a reasonable amount of comments. At the very beginning of each file, there should be a comment containing the course number, your name, the assignment number, and the date. It is good practice to insert comments into the source code, as you do when programming in any other languages.
```
/*
Class: ECE25100 Object Oriented Programming
Instructor: Xiaoli Yang
Author: [Your Name]
Assignment: [No.]
File Name:
Date: [MM]/[DD]/[YY]
*/
```
4. You should follow the programming styles and guidelines set forth in class. In particular, use meaningful names for variables. For each coding question you should also write a test class and test your code thoroughly.
5. Compress all .java files into one zip file and name it with your full name firstname_lastname_assignment #.zip. Submit the zip file on blackboard.

****

In this assignment you will simulate a database keeping track of students registered in courses at Purdue University Northwest. The purpose of the assignment is to get used to using ArrayLists as well as getting more experience in having objects interact together.

PART A: Implement a class called **Student** that has these **private** instance variables:

- **name** - a String indicating the name of the student.
- **year** - an int indicating the year that the student began his/her program.
- **studentNumber** - an int representing the student number

Implement the following instance methods:

- get methods for all instance variables
- A zero-argument constructor with reasonable default values
- A constructor that accepts as arguments the student's name, number and starting year.
- A **toString()** method that returns a string representation of the student like this: **Ralph Smith #100654321 (2018)**

PART B: Implement a class called **Course** that has these **private** instance variables:

- **title** - a String indicating the title of the course (e.g., "Object Oriented Programming").
- **code** - an String representing the department code for the course (e.g., "ECE");
- **number** - an int representing the course number (e.g., 25100);

Implement the following instance methods:

- get methods for all instance variables
- A zero-argument constructor with reasonable default values
- A constructor that accepts as arguments the course's code, number and title.
- A **toString()** method that returns a string representation of the course like this: **ECE25100 – Object Oriented Programming**

Test your code before continuing using the program below:

```
public class StudentCourseTestProgram {
    public static void main(String args[]) {
        Student s1 = new Student("Tommy Lauren", 100654321, 2016);
        Course c1 = new Course("BIOL", 10100, "Introductory Biology I");
        System.out.println(s1);
        System.out.println(c1);
    }
}
```

PART C: Now we will add functionality that allows students to register for courses.

- Add an instance variable to the **Student** class called **courses** which should maintain an **ArrayList<Course>** of courses that the student has registered in. Create a corresponding get method and initialize it to an empty list within your constructors.
- Create methods in the **Student** class called **addCourse(Course c)** and **removeCourse(Course c)** that add and remove courses from the Student's list of registered courses.
- Similarly, add an instance variable to the **Course** class called **students** which should maintain an **ArrayList<Student>** of students that are registered in the course. Create a corresponding get method and initialize it to an empty list within your constructors.
- Create methods in the **Course** class called **addStudent(Student s)** and **removeStudent(Student s)** that add and remove courses from the Student's list of registered courses.

Test your code before continuing using the program below:

```
public class StudentCourseTestProgram2 {
    public static void main(String args[]) {
        Student s1 = new Student("Jackie Chen", 100123456, 2015);
        Student s2 = new Student("Bruce Willis", 100234567, 2017);
        Student s3 = new Student("Keanu Reeves", 100345678, 2017);
        Course c1 = new Course("BIOL", 10100, "Introductory Biology I");
        Course c2 = new Course("BIOL", 21300," Human Anatomy and Physiology
I");
        Course c3 = new Course("BIOL", 22100, " Intro To Microbiology");
        s1.addCourse(c1);
        s1.addCourse(c2);
        s2.addCourse(c2);
        c1.addStudent(s1);
        c2.addStudent(s1);
        c2.addStudent(s2);
        System.out.println(s1.getCourses());  // should show 2 courses
        System.out.println(s2.getCourses());  // should show 1 course
        System.out.println(s3.getCourses());  // should show 0 courses
        System.out.println(c1.getStudents()); // should show 1 student
        System.out.println(c2.getStudents()); // should show 2 students
        System.out.println(c3.getStudents()); // should show 0 students
    }
}
```

PART D: Now implement a class called **University** that has these **private** instance variables:

- **name** - a String indicating the name of the University (e.g., "Purdue University Northwest").
- **courses** - an ArrayList<Course> representing the list of courses offered by the university.
- **students** - an ArrayList<Student> representing the list of students registered at the university.

Create the following public methods (see the testing code below for more on the method signatures):

- get methods for all three instance variables.
- a single constructor that takes the **name** of the university
- a method called **offerCourse(Course c)** which adds the given course to the list of courses that the University offers.
- a method called **cancelCourse(Course c)** which removes the given course from the list of courses that the University offers. The university must **not** allow a **Course** to be deleted that has students registered in it.
- a method called **enrollStudentInCourse(Student s, Course c)** which enrolls the given student into the given course. The university must **not** allow a **Student** to enroll for a course that does not exist.
- a method called **withdrawStudentFromCourse(Student s, Course c)** which withdraws the given student from the given course. The university must **not** allow a **Student** to withdraw from a course that he/she is not registered in.
- A **toString()** method that returns a string representation of the university like this (where the numbers will vary according to the sizes of the two array lists): **Purdue University Northwest: 26 Courses, 10 Students**

PART E: Now for the fun! Implement the following additional public instance methods in the **University** class:

- **lowRegistrationCourses(int min)** - returns an **ArrayList<Course>** of all courses that have less than or equal the **min** number of students enrolled in it.
- **highestEnrollment()** - returns the **Course** that has the maximum enrollment. In the case of a tie, return either course, but just one.
- **busiestStudent() -** returns the **Student** that is taking the most courses. In the case of a tie, return either student, but just one.

Create the following additional public instance method in the **Student** class:

- **classmatesAt(University u) -** takes a University **u** as a parameter and returns an **ArrayList<Student>** containing all students that are enrolled in at least one of the courses that this student is enrolled in. Note that your resulting array list should <u>not</u> contain the specified student, nor should it contain any duplicate student objects.

Here is the test case that you should run. Make sure that it works properly:

```
public class StudentCourseTestProgram {
    public static void main(String args[]) {
        Student s1 = new Student("Jackie Chen", 100123456, 2015);
        Student s2 = new Student("Bruce Willis", 100234567, 2017);
        Student s3 = new Student("Keanu Reeves", 100345678, 2017);
        Student s4 = new Student("Carrie Anne Moss", 100456789, 2015);
        Student s5 = new Student("Leonardo Dicaprio", 100567890, 2016);
        Student s6 = new Student("Cate Blanchett", 100678901, 2017);
        Student s7 = new Student("Clint Eastwood", 100789012, 2016);
        Student s8 = new Student("Hilary Swank", 100890123, 2017);
        Student s9 = new Student("Matt Damon", 100901234, 2011);
        Student s10 = new Student("Jack Nicholson", 100000001, 2017);

        Course c1 = new Course("BIOL", 10100, "Introductory Biology I");
        Course c2 = new Course("BIOL", 21300," Human Anatomy and Physiology
I");
        Course c3 = new Course("BIOL", 22100, " Intro To Microbiology");
        Course c4 = new Course("BIOL", 30700, "Plant Physiology");

        Course c5 = new Course("ECE",15200, "Programming for Engineers");
        Course c6 = new Course("ECE", 20100, "Linear Circuit Analysis I");
        Course c7 = new Course("ECE", 25100, "Object-Oriented Programming");
        Course c8 = new Course("ECE", 23300, "Micro Computers in
Engineering");
        Course c9 = new Course("ECE", 30200, "Probabilistic Methods in
Electrical Engineering");
        Course c10 = new Course("ECE", 38400, "Linear Control Systems");
        Course c11 = new Course("MGMT", 1001, "Principles of Financial
Accounting");
        Course c12 = new Course("MGMT", 50300, "Advanced Accounting");
        Course c13 = new Course("MGMT", 54400, "Database Management Systems");
        Course c14 = new Course("MGMT", 59000, "Directed Readings in
Management");
```

```
        Course c15 = new Course("CHM", 10000, "Preparation for General
Chemistry");
        Course c16 = new Course("CHM", 11500, "General Chemistry");
        Course c17 = new Course("CHM", 25600, "Organic Chemistry I");
        Course c18 = new Course("CHM", 49000, "Selected Topics in Chemistry
for Division Students");
        Course c19 = new Course("PSY", 12000, "Elementary Psychology");
        Course c20 = new Course("PSY", 20300, "Introduction to Research
Methods in Psychology");
        Course c21 = new Course("PSY", 20500, "Testing and Measurement");
        Course c22 = new Course("PSY", 31000, "Sensory and Perceptual
Processes");
        Course c23 = new Course("PSY", 31100, "Human Learning and Memory");
        Course c24 = new Course("PSY", 31400, "Introduction to Learning");
        Course c25 = new Course("PSY", 33900, "Advanced Social Psychology");
        Course c26 = new Course("PSY", 34400, "Human Sexuality");
        Course c27 = new Course("YUMMY", 1101, "Professional Sandwich
Making");

        University cu = new University("Purdue University Northwest");
        cu.offerCourse(c1);
        cu.offerCourse(c2);
        cu.offerCourse(c3);
        cu.offerCourse(c4);
        cu.offerCourse(c5);
        cu.offerCourse(c6);
        cu.offerCourse(c7);
        cu.offerCourse(c8);
        cu.offerCourse(c9);
        cu.offerCourse(c10);
        cu.offerCourse(c11);
        cu.offerCourse(c12);
        cu.offerCourse(c13);
        cu.offerCourse(c14);
        cu.offerCourse(c15);
        cu.offerCourse(c16);
        cu.offerCourse(c17);
        cu.offerCourse(c18);
        cu.offerCourse(c19);
        cu.offerCourse(c20);
        cu.offerCourse(c21);
        cu.offerCourse(c22);
        cu.offerCourse(c23);
        cu.offerCourse(c24);
        cu.offerCourse(c25);
        cu.offerCourse(c26);

        cu.enrollStudentInCourse(s1, c1);
        cu.enrollStudentInCourse(s1, c5);
        cu.enrollStudentInCourse(s1, c11);
        cu.enrollStudentInCourse(s1, c16);
        cu.enrollStudentInCourse(s2, c4);
        cu.enrollStudentInCourse(s2, c8);
        cu.enrollStudentInCourse(s2, c15);
        cu.enrollStudentInCourse(s3, c3);
        cu.enrollStudentInCourse(s3, c7);
        cu.enrollStudentInCourse(s3, c23);
        cu.enrollStudentInCourse(s3, c24);
        cu.enrollStudentInCourse(s3, c25);
        cu.enrollStudentInCourse(s4, c8);
        cu.enrollStudentInCourse(s4, c5);
        cu.enrollStudentInCourse(s4, c11);
        cu.enrollStudentInCourse(s4, c16);
        cu.enrollStudentInCourse(s5, c7);
        cu.enrollStudentInCourse(s5, c14);
        cu.enrollStudentInCourse(s5, c18);
        cu.enrollStudentInCourse(s5, c26);
        cu.enrollStudentInCourse(s6, c8);
        cu.enrollStudentInCourse(s6, c9);
        cu.enrollStudentInCourse(s6, c10);
        cu.enrollStudentInCourse(s6, c15);
        cu.enrollStudentInCourse(s7, c2);
        cu.enrollStudentInCourse(s7, c21);
        cu.enrollStudentInCourse(s7, c17);
        cu.enrollStudentInCourse(s7, c12);
        cu.enrollStudentInCourse(s7, c13);
        cu.enrollStudentInCourse(s7, c6);
        cu.enrollStudentInCourse(s8, c3);
        cu.enrollStudentInCourse(s8, c14);
        cu.enrollStudentInCourse(s8, c22);
        cu.enrollStudentInCourse(s9, c19);
        cu.enrollStudentInCourse(s9, c21);
        cu.enrollStudentInCourse(s10, c1);
        cu.enrollStudentInCourse(s1, c27);  // Should not work because course
is not offered
        cu.enrollStudentInCourse(s4, c11);  // This student is already
registered in this course, and should not be added twice
        cu.enrollStudentInCourse(s4, c11);  // Again this should not work

        System.out.println("The University looks as follows: " + cu);  // 10
students & 26 courses
         System.out.println("Attempting to cancel \" Introductory Biology I
\"...");
        cu.cancelCourse(c1);  // Should not work, but should not stop program
either
         System.out.println("The University still looks as follows: " + cu);
// 10 students & 26 courses
         System.out.println("Attempting to cancel \" Professional Sandwich
Making\"...");
        cu.cancelCourse(c27);  // Should not work, but should not stop
program either
        System.out.println("The University still looks as follows: " + cu);
// 10 students & 26 courses
         System.out.println("Attempting to cancel \"Introduction to Research
Methods in Psychology\"...");
        cu.cancelCourse(c20);  // Should work ok
        System.out.println("The University now looks as follows: " + cu); //
10 students & 25 courses
```

```
        System.out.println("Adding \" Introduction to Research Methods in 
Psychology\" again... ");
        cu.offerCourse(c20);

        System.out.println("\nThe courses with no students are: " + 
cu.lowRegistrationCourses(0));  // only c20

        System.out.println("\nThe courses with one student or less are: " + 
cu.lowRegistrationCourses(1));  // all these 
c2,c4,c6,c9,c10,c12,c13,c17,c18,c19,c20,c22,c23,c24,c25,c26

        System.out.println("\nThe course with the highest enrollment is " + 
cu.highestEnrollment()); // Micro Computers in Engineering

        System.out.println("The busiest student is " + cu.busiestStudent());

        System.out.println("\nThe classmates of Keanu Reeves are: " + 
s3.classmatesAt(cu));

        System.out.println("The classmates of Clint Eastwood are: " + 
s7.classmatesAt(cu));

        System.out.println("The University looks as follows: " + cu);  // 10 
students & 26 courses

        System.out.println("Clint Eastwood is taking these courses: " + 
s7.getCourses()); // should be 6 courses

        System.out.println("Withdrawing Clint Eastwood from \"Human Anatomy 
and Physiology I\" and from \"Linear Circuit Analysis I\"...");
        cu.withdrawStudentFromCourse(s7, c2);  // Should work ok
        cu.withdrawStudentFromCourse(s7, c6);  // Should work ok
        System.out.println("Clint Eastwood is now taking these courses: " + 
s7.getCourses()); // should be 4 courses

        System.out.println("\nThe busiest student now is " + 
cu.busiestStudent());// Keanu Reeves

        System.out.println("\nThe courses with no students are: " + 
cu.lowRegistrationCourses(0));  // now c2, c6 and c20
    }
}
```