CS 2510 Exam 2 – Summer 2010

Name: ______________________

Student Id (last 4 digits): ______________________

• Write down the answers in the space provided.

• You may use all parts of the Java language we have learned. If you need a method and you don't know whether it is provided, define it. You do not need to include the curly braces for every `if` or every `else`, as long as the statements you write are correct in standard Java.

• For tests you only need to provide the expression that computes the actual value, connecting it with an arrow to the expected value. For example `s.method() -> true` is sufficient.

• Remember that the phrase "develop a class" or "develop a method" means more than just providing a definition. It means to design them according to the **design recipe**. You are *not* required to provide a method template unless the problem specifically asks for one. However, be prepared to struggle if you choose to skip the template step.

• We will not answer *any* questions during the exam.

Problem	Points	/
1A		/ 4
1B		/22
Total		/26

Good luck.

26 Points

Problem 1

The registrar system at a university that keeps track of students enrolled in courses during the current semester consists of a list of courses and a list of students.

For each course we record the course name, the number of credits the course carries, and a roster of students registered for the course.

For each student we record student's name, id number, and the student's schedule — a list of courses student is enrolled in.

The class diagrams on the next two pages show the classes that represent the registrar system. The second alternative uses parametrized types for the lists. You can choose to work with either the first or the second alternative, but do not mix the two.

```
                    +-----------------------+
                    | Registrar             |
                    +-----------------------+
                +---| ILoC courses          |
                |   | ILoSt students        |---+
                |   +-----------------------+   |
                v                               v
           +---------+                     +---------+
           |  ILoC   |<------+             |  ILoSt  |<------+
           +---------+       |             +---------+       |
           +---------+       |             +---------+       |
                |            |                  |            |
                |            |                  |            |
               /_\           |                 /_\           |
                |            |                  |            |
      +--------------+       |        +--------------+       |
      v              v       |        v              v       |
+-------+  +--------------+  | +--------+  +---------------+ |
| MTLoC |  | ConsLoC      |  | | MTLoSt |  | ConsLoSt      | |
+-------+  +--------------+  | +--------+  +---------------+ |
         +-| Course first |  |           +-| Student first | |
         | | ILoC rest    |--+           | | ILoSt rest    |-+
         | +--------------+              | +---------------+
         v                               v
 +----------------+              +--------------+
 | Course         |              | Student      |
 +----------------+              +--------------+
 | String name    |    ^    ^    | String name  |
 | int credits    |    :    :    | int id       |
 | ILoSt students |----+    +----| ILoC courses |
 +----------------+              +--------------+
             +---------+
             |  ILo<T> |<------+
             +---------+       |
             +---------+       |
                  |            |
                  |            |
                 /_\           |
                  |            |
        +--------------+       |
        v              v       |
+---------+    +-------------+ |
| MTLo<T> |    | ConsLo<T>   | |
+---------+    +-------------+ |
           +---| T first     | |
           |   | ILo<T> rest |-+
           :   +-------------+
           v
```

```
+---------------------+
| Student             |
+---------------------+
| String name         |
| int id              |
| ILo<Course> courses |---->
+---------------------+
```

```
+-----------------------+
| Course                |
+-----------------------+
| String name           |
| int credits           |
| ILo<Student> students |---->
+-----------------------+
```

```
+-----------------------+
| Registrar             |
+-----------------------+
| ILo<Course> courses   |---->
| ILo<Student> students |---->
+-----------------------+
```

A. 4 POINTS

The examples given define the initial state of the registrar system before any students registered for courses.

```
  // sample students
  Student jan = new Student("Jan", 123);
  Student dan = new Student("Dan", 234);
  Student kim = new Student("Kim", 345);
  Student pat = new Student("Pat", 567);

  // sample courses
  Course math = new Course("Math", 4);
  Course chem = new Course("Chem", 4);
  Course engl = new Course("Engl", 4);
  Course hist = new Course("Hist", 4);
  Course phys = new Course("Phys", 4);
  Course comp = new Course("Comp", 4);
  Course band = new Course("Band", 2);
  Course swim = new Course("Swim", 2);
  Course draw = new Course("Draw", 2);

  // this really should be a sequence of 'new ConsLoC('
  // ending with 'new MtLoC()'
  ILoC allcourses =
    ...(math, chem, engl, hist, phys, comp,
        band, swim, draw)...;

  // this really should be a sequence of 'new ConsLoS('
  // ending with 'new MtLoS()'
  ILoS allstudents =
    ...(jan, dan, kim, pat)...;

/*-----------------------------------------------------
-- Alternate parametrized version:
   use ILo<Courses>, ConsLo<Courses>, MtLo<Courses
   and also ILo<Student>, ConsLo<Student, MtLo<Student>
-----------------------------------------------------*/

  Registrar reg = new Registrar(allcourses,
                                allstudents);
```

Suppose the method `register` allows a student to register for a course. It consumes the student name and id and the course name and makes the necessary changes to enroll student in a course. Show the state of the registrar system after the following registration actions have taken place:

```
this.reg.register("Kim", 345, "Math");
this.reg.register("Kim",345, "Engl");
this.reg.register("Kim",345, "Chem");
this.reg.register("Kim",345, "Hist");
this.reg.register("Kim",345, "Draw");

this.reg.register("Jan", 123, "Math");
this.reg.register("Jan",123, "Hist");

this.reg.register("Pat",567, "Engl");

this.reg.register("Dan",234, "Chem");
this.reg.register("Dan",234, "Draw");
```

Course name	Credits	Students registered
math	4	kim, jan

Student name	Id	Credits	Current course schedule
Dan	234	6	chem, draw

__________**Solution** __________ [POINTS 4: 2 points for courses, 2 points for students]

```
// to represent a lits of items of the type T
interface ILo<T>{}
...
```

B. 22 POINTS

Design the method `register` for the class `Registrar`.

The method should make sure the following conditions have been met and should throw an exception if the registration cannot be completed, indicating the reason why the student could not be registered for the course.

- The student must appear in the registrar's list of students
- The course must appear in the registrar's list of courses
- A student may not register for more than 16 credits
- A student may not register for the same course twice

Note: You will need several helper methods here. Design each of them carefully. You may want to outline your strategy in English. You will get credit for well designed helper methods.

Note: The *DESIGN RECIPE* is your friend here.

... This page is intentionally blank — for your work ...

... This page is intentionally blank — for your work ...

______________**Solution** ______________ [POINTS 20: one point for the `SimpleGift`, and one point for the example with two `Boxes`.]

```
SoccerLeague sl = new SoccerLeague();
```