**LEBANESEAMERICANUNIVERSITY**

**School of Arts and Sciences**

**Department of Computer Science and Mathematics**

Fall 2013

**Assignment 6 (Exceptions and Files)**

**Due on Tuesday December 17, 2013**

**Exercise I:**

|  |
| --- |
| **RegistrationCheck** |
|  |
| + registerStudent(student : Student, courseId : int) : void |

|  |
| --- |
| **Student** |
| - id : long- name : String- coursesFinished : Course[] |
| + Student(id : long, name : String, coursesFinished : Course[])+ setId(id : long) : void+ getId() : long+ setName(name : String) : void+ getName(): String+ setCoursesFinished(coursesFinished : Course[]) : void+ getCoursesFinished() : Course[]+ toString() : String |

|  |
| --- |
| **Course** |
| - id : int- name : String- preRequisite : Course |
| + Course(id : int, name : String, preRequisites: Course[])+ setId(id : int) : void+ getId() : int+ setName(name : String) : void+ getName(): String+ setPreRequisite(preRequisite: Course) : void+ getPreRequisite() : Course+ toString() : String |

|  |
| --- |
| **RegistrationCheckSystem** |
| - courses : Course[] |
| + registerStudent(student : Student, courseId : int) : void |

|  |
| --- |
| **RegistrationException** |
|  |
| + RegistrationException(message : String) |

|  |
| --- |
| **CourseNotFoundException** |
|  |
| + CourseNotFoundException(message : String) |

In this assignment, you will be creating a Course Registration Checking System which will check if a student can register a course or not.

First, you have an interface which is the Registration interface that has one method declaration. The method is registerStudent which takes a student and a course ID as parameters.

The class Student has three private variables which are id, name, and courses finished. These variables should be initialized in the constructor and should have set and get methods.

The Course class has three private variables which are id, name, and pre-requisites. These variables should be initialized in the constructor and should have set and get methods.

The RegistrationSystem class has an array of courses and should implement the registration interface. This means that this class should implement the registerStudent method. The class has an array of available courses.

The registerStudent method should check if the student can register a specific course. The method checks if the student has finished the pre-requisites of a specific course. If so, the methods should print that the student can register this course. If the student has not completed the pre-requisites, a RegistrationException should be thrown. If the course that the student wants to register is not found, a CourseNotFoundException should be thrown.

The CourseNotFoundExceptionclass should extends the Java Exception class and call the superclass’ constructor in its constructor.

The RegistrationException class should extends the Java Exception class and call the superclass’ constructor in its constructor.

Write a tester class that creates a student and checks if a student can register a course. For simplicity, limit the number of courses to five courses.

Input should be read from the user via a scanner. First read the user id, name, and the courses he or she has finished. The number of finished courses should be two courses only.

Then read an integer which will be the course id of the course that the student wants to register. Then call resgiterStudent passing the student and the course id as a parameter.

There are a number of exceptions that should be handled in the tester class.

InpuMissmatchException might be thrown if the user enters invalid input. If so, you should tell the user that her or his input is wrong and keep requesting to enter a valid input.

The other exception is CourseNotFoundException that will be thrown if the user enters a course id that is not found. You should handle this exception by keeping on requesting from the user to enter a valid course id.

RegistrationException will be thrown if the student has not completed the prerequisites of the course. This should be handled by printing that the student cannot register this course.

**Hints:**

* Use a while loop to keep requesting valid input from the user.
* Use the following table as a course description:

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **preRequisite** |
| 10 | CSC243 | - |
| 11 | CSC245 | CSC243 |
| 12 | CSC310 | CSC245 |
| 13 | MTH201 | - |
| 14 | MTH202 | MTH201 |

**Sample Run:**

Student id: 20110515

Student name: Test Student

Courses finished: 10 13

Course to register: test

Please enter a valid input!

Course to register: 12

Student has not completed the course prerequisites

**Exercise II:**

To solve this assignment, you are supposed to know file input/output (reading from and writing to a file) and array list usage. **You will present your solution as a demo during the next lab on Friday**.

 First, fill an input file called *input.txt* with 12 numbers: **4 6 8 3 1 2 3 9 8 4 3 0 (each integer on a line)**.

 In your main method, read these items from the input file, fill them in an integer array list, and write them to an output file called *output.txt* after concatenating each of them with the corresponding index.

The content of the output file will be: **4 0**

**6 1**

**8 2**

**3 3**

**1 4**

**2 5**

**3 6**

**9 7**

**8 8**

**4 9**

**3 10**

**0 11**

**Deliverables:**

You must submit an electronic copy of this assignment (zip files containing only the java files) through the blackboard to your lab instructors. You have also to do a demo during the first lab on or after the due date. Lab Instructor Email: mohammad.elmedawar@gmail.com ).

**Policy on Cheating and Plagiarism:**

Plagiarism on assignments and project work is a serious offense. If plagiarism is detected, a student will be subject to penalty, which ranges from receiving a zero on the assignment concerned to an “F” in the course.