PROGRAMMING ASSIGNMENT 1

Due Date : 18.03.2015

Introduction: The aim of this experiment is to introduce object oriented programming to students by the Java programming language. By the help of this experiment students will learn the structure of a class, how classes interact, how multiple instances of classes are controlled and basic input-output operations in Java.

EXPERIMENT

In this experiment, you are expected to develop a simple Student Registration System. In this system there are students and courses. Students and courses have an id number. Students can register any course they want depending on these constraints;

1. In this system there will be maximum **40 courses and 400 students**.
2. Students can register up to **8 courses or 22 credits**
3. Each course can support certain number of student depending on it's quota.
4. Student and course ids must be unique.

In addition, the system will have a menu and support these operations;

1. Adding Operations
   1.1. Add a student
   1.2. Add a student to a specified course
   1.3. Add a course
2. Deleting Operations
   2.1. Delete a student by student id
   2.2. Delete a student by student id from a specified course
   2.3. Delete a course by course id
3. Listing Operations
   3.1. List all students who registered a specified course
   3.2. List all courses that a student registered
   3.3. List all students
   3.4. List all courses
4. Exit

An example for input-output format has been given below;

```
1.Adding operations
2.Deleting operations
3.Listing operations
4.Exit

1
1.Add a student
2.Add a student to a specified course
3.Add a course
4.Return to main menu

1
Enter Student Id: 223543
Enter Name: Ayse
Enter Surname: Sarikaya
Done!

1.Add a student
2.Add a student to a specified course
3.Add a course
4.Return to main menu

2
Enter Student Id: 223543
Enter Course Id: Bil131
Done!

1.Add a student
2.Add a student to a specified course
3.Add a course
4.Return to main menu

3
Enter Course Id: Bil131
Enter Name: Computer_Programming_I
Enter Credit: 3
Enter Quota: 30
Done!

1.Add a student
2.Add a student to a specified course
3.Add a course
4.Return to main menu

4
1.Adding operations
```

```
2.Deleting operations
3.Listing operations
4.Exit

2
1.Delete a student by student id
2.Delete a student by student id from a specified course
3.Delete a course by course id
4.Return to main menu

1
Enter Student Id: 223543
Done!

1.Delete a student by student id
2.Delete a student by student id from a specified course
3.Delete a course by course id
4.Return to main menu

2
Enter Student Id: 223543
Enter Course Id: Bil131
Done!

1.Delete a student by student id
2.Delete a student by student id from a specified course
3.Delete a course by course id
4.Return to main menu

3
Enter Course Id: Bil131
Done!

1.Delete a student by student id
2.Delete a student by student id from a specified course
3.Delete a course by course id
4.Return to main menu

4
1.Adding operations
2.Deleting operations
3.Listing operations
4.Exit

3
1.List all students who registered a specified course
2.List all courses that a student registered
```

```
3.List all students
4.List all courses
5.Return to main menu

1
Enter Course Id: Bil131
Computer_Programming_I 3 30
223543 Ayse Sarikaya
324211 Damla Yazici
326601 Hulya Ulugut

1.List all students who registered a specified course
2.List all courses that a student registered
3.List all students
4.List all courses
5.Return to main menu

2
Enter Student Id: 324211
Damla Yazici
Bil131 Computer_Programming_I 3 30
Bil138 Computer_Lab_II 1 20
Bil137 Programming_Lab_II 2 25

1.List all students who registered a specified course
2.List all courses that a student registered
3.List all students
4.List all courses
5.Return to main menu

3
223543 Ayse Sarikaya
324211 Damla Yazici
326601 Hulya Ulugut
348983 Meltem Demiray
902321 Ferhat Kuvvetoglu
881291 Emel Saritas
891232 Hasan Ercelebi
122323 Ali Kilicarslan

1.List all students who registered a specified course
2.List all courses that a student registered
3.List all students
4.List all courses
5.Return to main menu

4
```

```
Bil131 Computer_Programming_I 3 30
Bil137 Programming_Lab_I 2 25
Bil132 Computer_Programming_II 3 40
Bil138 Computer_Lab_II 1 20

1.List all students who registered a specified course
2.List all courses that a student registered
3.List all students
4.List all courses
5.Return to main menu

5
1.Adding operations
2.Deleting operations
3.Listing operations
4.Exit

4
```

Error Handling: You must also handle the situations such as;

1. Whether there is a specified course for adding a student or for deleting
2. Whether there is a student for registration process or for deleting
3. Whether there is a same student id or course id before adding student or course
4. Whether operations comply with the constraints mentioned above.

Notes

- Don't miss the deadline.
- Save all your work until the assignment is graded.
- The assignment must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- You can ask your questions via Piazza (`https://piazza.com/hacettepe.edu.tr/spring2015/bbm104`) and you are supposed to be aware of everything discussed in Piazza.
- Save all your work until the assignment is graded.

This file hierarchy must be zipped before submitted (Not .rar , only .zip files are supported by the system). Additionally, your "public static void main" method must be included in Main.java

→ <student id>
→ Main.java
→ *.java

Late Policy: You may use up to three extension days for the assignment. But each extension day will bring about additional 10% degradation for evaluation of the assignment.