The Hong Kong Polytechnic University
Department of Electronic and Information Engineering

EIE3320 Tutorial 3: Object-Oriented Programming (Grouping Objects)

(Deadline for Submission: Check the course information)

1. **(Assignment)** Create two classes `Course` and `UniversityStudent`.

The class `Course` is to represent a course. It has two data fields and two methods.

- Data field `courseName`: It is a string to store the name of the course.
- Data field `testMarks`: It is an integer to store the result of the course.
- Constructor: It gets the name and the result of the course when an object is created.
- Method `toString`: It returns a string with the name and the result of the course. Note that arguments should be passed to this method.

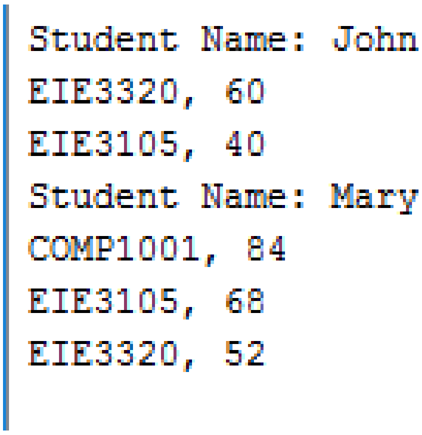
The class `UniversityStudent` is to represent a university student. It has three data fields and two methods.

- Data field `studentName`: It is a string to store the student's name.
- Data field `courseNumber`: It is an integer to store the number of courses that the student takes.
- Constructor: It gets the student's name, the number of courses that the student takes and an array of objects from the class `Course` when an object is created. The array size should be 10.
- Method print: It displays the student's name and the information of all courses that the student takes. Note that arguments should be passed to this method. Moreover, you **MUST** invoke the method `toString` in objects from the class `Course`.

The testing program is shown below:

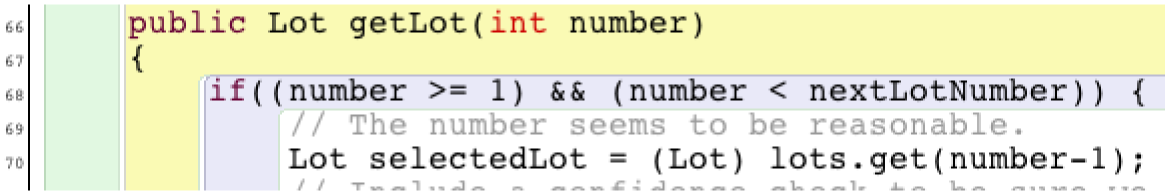
```
public class Test1Q2A
{
    public static void main(String[] args)
    {
        Course[] listA = new Course[10];
        listA[0] = new Course("EIE3320", 60);
        listA[1] = new Course("EIE3105", 40);
        UniversityStudent studentA = new UniversityStudent("John", 2,
listA);
        studentA.print();
        Course[] listB = new Course[10];
        listB[0] = new Course("COMP1001", 84);
        listB[1] = new Course("EIE3105", 68);
        listB[2] = new Course("EIE3320", 52);
        UniversityStudent studentB = new UniversityStudent("Mary", 3,
listB);
        studentB.print();

    }
}
```

The output of the testing program is shown below for your reference.



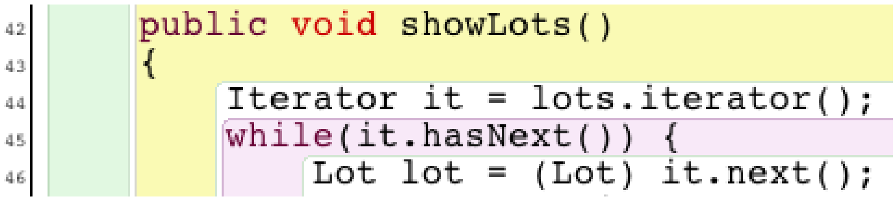
```
Student Name: John
EIE3320, 60
EIE3105, 40
Student Name: Mary
COMP1001, 84
EIE3105, 68
EIE3320, 52
```

2. Turn on Line Number Display by selecting Tool → Preferences → Display Line Numbers. Find examples of **casting** in the `Auction` class. Starting from Java 5, casting is not necessary for the `get()` method. Try removing the casting in Line 70 of "Aution.java", i.e., removing `(Lot)` in



```
66  public Lot getLot(int number)
67  {
68      if((number >= 1) && (number < nextLotNumber)) {
69          // The number seems to be reasonable.
70          Lot selectedLot = (Lot) lots.get(number-1);
```

Then, recompile "Auction.java". Read the Java API (http://docs.oracle.com/javase/6/docs/api/) to see why casting is not necessary. Now, remove the casting in Line 46 of "Aution.java".



```
42  public void showLots()
43  {
44      Iterator it = lots.iterator();
45      while(it.hasNext()) {
46          Lot lot = (Lot) it.next();
```

Recompile the program and you will see that it fails to compile. Modify Line 44 to solve the problem.

Lawrence Cheung
August 2017