CST8132 Object-Oriented Programming

Lab 2: College System I

Due Date: Week 3 – in own lab hours

Marks: 10 marks (worth 4% of term mark)

Demo: Demo your code and output to your lab professor during your own lab hours.

Recommended Reading: Chapter 9 of Deitel and Deitel, Java How to Program book

Exercise

We will create a few classes to represent a college system. A college has students registered with them. Each student has attributes of a person (first name, last name, email, and phone number) and a gpa, which is calculated based on the marks obtained in various courses (number of courses can vary). We will then add faculty also to the system. More classes and properties will be added later.

As the first step, you need to create the following classes:

`Person` class

Instance variables (private): `firstName`(String), `lastName` (String), `emailId` (String), `phoneNumber` (long)

Constructor: parameterized constructor that gets values to set all properties of a person

Methods: getters to return name, email ID and phone number. Name should be returned as one string. If first name is "John" and last name is "Doe", getter should return "John Doe".

Student class

Instance variables: `studentNumber` (int), `student` (Person), `gpa` (double)

Constructor: parameterized constructor that initializes a student with student number and all personal properties. (do you need a no-arg constructor??? Think and decide!)

Methods:

1. `readStudentInfo()`: accepts nothing, return nothing. Reads all student information. You can declare local variables to read information, and then create student by invoking the parameterized constructor of Person class (student = new Person(…….);). In order to read marks, `readMarks()` method will be called from this method.
2. `readMarks()`: accepts nothing, return nothing. Reads number of courses, and then reads marks of all courses and stored them in a local double array. After reading the marks, this method will call `calculateGpa()`.

3. `calculateGpa()`: accepts a double array, returns nothing. This method is always called from within the class (private method???). This method calculates the gpa and store it in the instance variable gpa.
4. `printStudent()`: accepts nothing, returns nothing. This method prints details of a student using formatted output (printf).

CollegeSystemTest class

This is the driver class (test class), which means this class has the main method.

Main method

- This method read the number of students (store it in num), and then create an array of Student names students (Student []students = new Student[num]).
- In a for loop, read details of students by calling readStudentInfo() method. (even though you create the students[] array, each object in the array need to be created in the for loop).
- Prints the title and the header row
- In a for loop, call printStudent() method to print details of all students

Format your code with proper indentation and other formatting. Your code should be properly commented. Test plan and external documentation are not required for this exercise, but in future labs they will be required.

Grading Scheme

Item	Marks
Person class (correct access specifiers, constructors, 3 getters)	2
Student class (correct access specifiers, constructors, 4 methods)	3
CollegeSystemTest class	2
Comments (class header, provide comments wherever required)	3

Submission

Zip your code in a folder named <LastName><FirstNameLab2.zip and submit it to Brightspace before the due date. Demonstrate your work to your lab professor during your own lab hours. Both submission and demo are required to get grades.

Getting ready for next lab

Once you are done with this lab, think how you can do this lab with inheritance – by extending Person class (i.e., instead of having student in Student class, extend Person class so that you can use all attributes of Person class in Student class. You need to think about the access specifiers also).

Expected Output (user input in blue)

```
Enter number of students: 3
Enter details of student 1
*************************
Enter student number: 1002
Enter first name: John
Enter last name: Doe
Enter email Id: doe@test.com
Enter phone number: 123456789
Enter number of courses: 3
Enter mark 1: 98
Enter mark 2: 96
Enter mark 3: 97
Enter details of student 2
*************************
Enter student number: 1005
Enter first name: Mark
Enter last name: Williams
Enter email Id: mark@test.com
Enter phone number: 213546879
Enter number of courses: 2
Enter mark 1: 100
Enter mark 2: 100
Enter details of student 3
*************************
Enter student number: 1008
Enter first name: Paul
Enter last name: Webster
Enter email Id: paul@test.com
Enter phone number: 321654987
Enter number of courses: 4
Enter mark 1: 86
Enter mark 2: 95
Enter mark 3: 87
Enter mark 4: 98

List of Students
******************
Student#         Name            Email           Phone    GPA
 1002 |            John Doe |    doe@test.com |    123456789 | 3.88 |
 1005 |        Mark Williams |   mark@test.com |    213546879 | 4.00 |
 1008 |         Paul Webster |   paul@test.com |    321654987 | 3.66 |
```