Langara
THE COLLEGE OF HIGHER LEARNING.

Department of Computing Science & Information Systems
CPSC 1181

Lab#1
Sep 15, 2022

Objectives:

- Learn to design, implement, and document a class
- Design test class

Preparation:

Study class notes and sections 3.1, 3.2, 3.3, 3.6, and 3.7 from your text book.

Due date:

Due Date: 11:00 PM on Wednesday, Sep 21, 2022.

Where to upload:

Zip your files into yourstudentID.zip where yourstudentID is your student number, and upload it to dropbox lab1 in D2L.

Part A: Tutorial

Do Tutorial1
You should finish tutorial and upload your files by 11:00 PM, Sep 15, 2022

Part B: Lab Assignment

Design and implement a class called Student

- A student has a name and a surname
- A student has a student number
- A student has an address
- A student has a quiz point average
- A student has a login id that is assigned ‘by the system’ when the student is constructed (you do not need to check for uniqueness of the login id – just follow the rules on assigning the login id)

Create appropriate constructors, instance fields, public/private methods for the Student class. Provide the following public methods:

Constructors:

```
// Constructor
Student(String firstName, String surName, long studentNumber){}
```

Accessors and mutator methods:

```
// set student's name and surname. Changing student's name does not affect the
students's loginId
void setName(String firstName, String surName)

// returns name and surname separated by comma (name, surname)
String getName();


// returns student number
long getStudentNumber();


// returns the students login Id. Login Id is generated by class Student.
// Refer to “About Your Implementation” for details.
String getLoginId();


// Returns student's info with following format
// name, surname (Student login Id, student number)
// Check sample test case for format
String getInfo();


// sets address of the student
void setAddress(String number, String street, String city, String province,
                                                           String postalCode) ;

// returns student's address
String getAddress();


// Add a quiz score to the student
void addQuiz(double quiz);


// Returns the average of the quiz of a student
double getQuizAverage();


// override toString() method
String toString();
```

Notes:

- The object does not need to access and modify quiz score.
- Assume all the quiz scores are out of 10.
- A student may have many quiz.
- Use appropriate keywords to encapsulate your design.

Part C: Lab Assignment

Address is an object by itself.

Implement `class Address` with following instance fields:

number, street, city, province, postal code

Provide appropriate constructors and methods for class Address, and use Address as an object in Student class.

Add appropriate methods to the class Address. You are free to design Address class based on your preference.

About your implementation:

1. Use Javadoc notation to comment all classes and methods.

2. Generating login id for students:

- the first letter of the 'login id' is the first letter of the student's name
- the middle part of the 'login id' consists of the first 4 letters of student' family name
- the last part of the 'login id' consists of the last 2 digits of the 'student number' i.e. the two least significant digits
- if the surname has fewer than 4 characters, then use all the letters of the surname
- the letters of the 'login id' are all lower case
- For example if student info is Hossein Darbandi (10003456), then the login Id is set to hdarb56

3. Implement class StudentTester.java and test your classes. The purpose of a test program (test harness) is to verify that the methods of the class have been implemented correctly. A test program creates objects of the class, and then calls their methods. Test engineers compare the values returned by the methods with expected values to verify the test is passed or failed.

4. Run StudentTester.java and capture output of each test cases. Document the test cases properly in testCase.docx file and verify that all the tests cases passed.

Check the sample StudentTester.java and and testCase.docx files.

Note that your marker can only access word files.

What to submit:

- Zip Student.java , Address.java, StudentTester.java, and testCase.docx files into yourStudentId.zip file and submit it to dropbox lab1 in D2L.
- testCase.docx file is a word file format. Other file formats may not be accepted by your marker.
- comments about your assignment if needed. These comments are not the comments documenting your code but rather something you need to convey to your marker about your assignment .
- If you have not finished your assignment, add comment why it is incomplete, what is working, what is not working, and why.

TOTAL MARK: 50