CPE 111 – Programming with Data Structures
Sections C and D - January 2021
Laboratory Exercise 13

Objective

This lab is intended to give you practice using ***ArrayList***, one of the simplest of the Java collection classes.

Instructions
Summary

Write a Java application called **StudentManager** that creates and manipulates a list of students which are instances of the ***Student*** class.

Details

1. Download the two Java source files **IOUtils.java** and **Student.java**. You will use these classes as part of your application. Look at the code for **Student.java**. You will see that this is a simple data class with a few methods. Notice that the constructor asks the user for the information to store in the class instance.

2. Create a Java source file called **StudentManager.java** which defines a public class called ***StudentManager***. Be sure to import the class ***java.util.ArrayList***.

3. At the top of the class, create a protected data member which is an ***ArrayList*** of ***Students***. For example:

```
protected static ArrayList<Student> allStudents = new ArrayList<Student>();
```

4. Create a static method called *getStudents()*. This method should loop, creating students by calling the *Student()* constructor, and then adding each **Student** to the *allStudents* list. The logic should be as follows:

```
bContinue = true
  while (bContinue)
     create student
     add student to list
     ask if user wants to add more
     if answer is no
           set bContinue to false
     endif
 end while
```

5. Create a static method called *listStudents*(). This method should iterate through the ***ArrayList***, printing only the *name of each student*. You can use an iterator or a for loop. To print the name, you can use code like the following:

```
Student nextStudent;
.....   /* get the next student from the array list */
System.out.println(nextStudent.getStudentName());
```

6. Create a *main()* method. Inside the *main()*, call *getStudents()* and then *listStudents()*.

7. Compile and test your class.

8. Now add a method called *findStudents()*. This method will ask for a student's ID, then look in the ***ArrayList*** to see if any student with that ID exists. If the student is found, the method should call the *display()* method on that student and then the *computeAvgGrade()* method and print the results. If the matching student is not found, the method should print an error message. The method should loop, asking for new ID values, until the user enters a value of 0. You should use the *IOUtils.getInteger()* method to get the ID value.

9. In *main()*, after the calls to *getStudents()* and *listStudents(),* add a call to *findStudents().* Compile and test your program. It should now go through three phases: asking for students, printing all student names, then asking for IDs and printing student information if found.

10. Upload **StudentManager.java**.

11. Extra credit: replace your worst lab score with the score for this lab. Copy **StudentManager.java** to **StudentManagerHash.java**. Change the implementation to use the ***Hashtable*** class instead of the call *getStudents()* and then *listStudents()* You should use the student ID as the hash key and the **Student** instance as the hash value. You will have to change the *int* to an ***Integer*** class instance to do this:

```
Integer key = new Integer(Student.getStudentId());
```

Compile and test. The behavior should not be any different from **StudentManager** but the code will be shorter. Also, if you had many students, the hash table implementation would be faster. Upload **StudentManagerHash.java.**