FIT9131 Programming Foundations

Week 6 Exercises

A. Homework checklist

To be up to date you should have completed the following:

- Lab exercises from weeks 1– 5.
- Read Chapters 1–4, sections 6.3-6.5, 6.11, 6.12 from Chapter 6, and sections 7.1-7.4 from Chapter 7, of the text book, *Objects First with Java*, Barnes & Kölling.

Assignment 1 checklist:

Your assignment should be well underway. By the end of this week you should have all or most of the coding completed.

Reminders:

- You MUST ATTEND your lab next week, WEEK 7, to organise your interview for Assignment 1 with your own tutor. Please remember, if no interview is scheduled, no marks will be awarded for your assignment.
- You should be showing you tutor your code (during the lab or helpdesk sessions) for the assignment that you have completed so far to get some feedback on what you have done.
- Assignment 1 is due in Week 7. There is no extension to the due date.

B. Exercises for Week 6

Create a new project in BlueJ for your work this week. Name the project **`Week6`** and create all your classes for the week in this project.

Have the current version of your Assignment 1 ready to show your tutor during the class
Your tutor will make a note of your progress and give you some feedback on what you have done.

1. Working with loops

This exercise will give you practice in using loops. Looping/iteration/repetition is a fundamental construct which allows tasks to be performed repeatedly until a specific condition is met.

Java supports the following types of loops:

- Pre-Test loops
  - **`for`** loop
  - **`while`** loop
  - **`for-each`** loop (for use with collections)
- Post-Test loop
  - **`do...while`** loop

When working with the **`for`**, **`while`** and **`do...while`** loop, it is important to remember that each of these loops is made up of the following components:

- Initialization counter / flag
- Boolean exit condition
- Increment counter / reset flag

(Note this does not apply to the **`for-each`** loop, which is used only with collections.)

We will illustrate these components in code examples:

`while` Loop:

```
int counter = 0;
while (counter < 5)
{
    System.out.println(counter);        //Body
    counter++;
}
```

`do` … `while` Loop:

```
int counter = 0;
do
{
    System.out.println(counter);        //Body
    counter++;
} while (counter <= 5);
```

`for` Loop:

```
for (int counter = 0; counter < 5; counter++)
{
    System.out.println(counter);        //Body
}
```

Note that all loops (except **`for-each`**) include ALL THREE components. The only thing which changes are the positions of *WHERE* they appear in the code.

`for-each` loop example:

`for-each` Loop:

```
for (Student aStudent : students)
{
    System.out.println(aStudent.display());
}
```

students is a collection of variable or fixed size (Arraylist or Array).

aStudent is a single object of the same type of object (Student) stored in the collection.

display() is a method which has been defined in the Student class accessible via the object aStudent.

a. Create a class in BlueJ called **`Week6Q1`** (in the **`Week6`** project), then add a method in it called **`displayMenu()`** that prints the following menu to the screen and asks the user to choose one of the options presented, as follows:

```
A. Option #A
B. Option #B
C. Option #C
D. Option #D
X. Exit!
Please enter your choice:
```

The method should do nothing else other than printing the menu.

b. Add a method to the **`Week6Q1`** class in Q.1, and name it **`start()`**. In this method, make a ***single*** call to **`displayMenu()`**, then use a **`Scanner`** object to accept an input from the user. When the user enters an option (**`A/B/C/D/X`**) from the menu, in upper or lower case, the program should print an appropriate message indicating which option has been selected (e.g. "**`Option #A was selected`**"). There is no need to actually perform any real action, apart from the printed message.

If the user enters an option other than those on the menu, an error message should be printed.

Hint: you need to use the **`if`**/**`switch`** statements in your code.

There is no need to perform any input validations.

Call the **`start()`** method several times (via *BlueJ*) to check that you have it working as required.

c. Create another method and name it **`startWhile().`**

This code for this method would be a modification of the original **`start()`** method. In this new version, add a **`while`** loop so that the menu is displayed repeatedly, until the last option is selected. When the last option is selected, exit the method (i.e. stop the loop).

Hint: you need to use a combination of the **`if`**/**`switch`** and **`while`** statements in your code.

2. Arrays

The following example will give you practice in using the **`Array`** class.

a. Write a *declaration* for an array variable called **`vacant`** that could be used to refer to an array of **`boolean`** values.

b. Write a *declaration* for an array variable called **`people`** that could be used to refer to an array of **`Person`** objects.

c. Given the following declaration:

```
Book[] library;
```

Write code to make the **`library`** variable refer to an array that can hold 100 **`Book`** objects.

d. How many **`String`** objects are created by the following declaration?

```
String[] labels = new String[20];
```

3. Conditional operator

Rewrite the following code so that the boolean value assigned to the **`concession`** variable is performed using the ***conditional operator***, instead of the **`if-else`** statement.

```
boolean concession = true;
char status = 'P';
if (status == 'p' || status == 'P')
    concession = true;
else
    concession = false;
```

C. Homework

1. Finish the lab exercises for week 6.

2. Read sections 9.1-9.2, 9.3.2, 9.4.1, 9.6, 9.8, 9.9.

3. ***Continue working on the assignment***.

D. Pre-lab tasks to be assessed in Week 7

1. Write a method that is passed an array of 12 monthly rainfall values of type **`double`** and displays on the screen the average **monthly** rainfall.

2. Open the **`music-organizer-v1`** project from the chapter04 projects folder. Using an **`Iterator`**, add a method to the **`MusicOrganiser`** class called **`displayCollectionWeek7`** that will display on the screen the names of the audio files in the collection. Display each name on a separate line. At the end of the list display the total number of files in the collection. Note: last week this method was written using the **`for-each`** loop and a **`while`** loop.