COMP1102/8702 – Practical Class 3

Decisions

For this lab exercise we will begin to look at making decisions within our code. Selection or decisions are a key component of any program that you write. By allowing the program to make decisions based on values provided not only allows for interactivity but also enables a generalised set of instructions that could be applied to many variations of execution.

Simple Flow of Control

if Statement, *if . . . else* Statement

In the programs you have seen so far, we have had a list of statements, which were executed in the order that they were written in your program (the `.java` file). In more complicated programs, you may need to change the order in which statements are executed. The order of execution for statements in your program is referred to as **flow of control**.

Let's look at one example. Suppose you are organising an event that costs $12 for everyone older than 8 and $6 for any one 8 years or younger. One way to do this is to say the ticket is $12, unless you are 8 or younger, then it is $6. In this case, you can write:

pseudo code
ticket = 12 age if age is less than or equal to 8 then ticket = 6

java equivalent

double ticket = 12.00;
double age;
if(**age <= 8**)
`ticket = 6;`

There is another way to do this.

pseudo code

if age is less than or equal to 8 then
ticket = 6
else
ticket = 12

java equivalent

if(**age <= 8**)
`ticket = 6;`
else
`ticket = 12;`

Both of these do the same thing. In both cases, you will change the flow of execution when you reach the statement; "**age is 8 or younger**". If that statement happens to be true, i.e., age is 8 or younger, then the value of ticket will change to $6, otherwise, you will go with its initialised value of $12.

In general, the statement in the parentheses is either **TRUE** or **FALSE**. Depending on that being true or false, you will change the flow of execution of the statements in the program. In order for your program to decide about the flow of the execution, it uses a comparison operator.

Examples of comparison operators are:

1) *equal to* , = , which will be written as == (2 =’s) in java, with a general form of:
statement1 == statement2.
Example: *y == x + 1*

2) *not equal to* ,≠, which will be written as *!=* in java, with a general form of:
statement1 != statement2.
Example: *y != x + 1*

3) *less than* , <, which will be written as < in java, with a general form of:
statement1 < statement2.
Example: *y < x + 1*

4) *less than or equal to* ,≤, which will be written as <= in java, with a general form of:
statement1 <= statement2.
Example: *y <= x + 1*

5) *greater than* , >, which will be written as > in java, with a general form of:
statement1 > statement2.
Example: *y > x + 1*

6) *greater than or equal to* ,≥, which will be written as >= in java, with a general form of:
statement1 >= statement2.
Example: *y >= x + 1*

7) *OR*, which will be written as || (2 of the |’s) in java, with a general form of:
statement1 || statement2.
Which may be True when EITHER one of the two statements are TRUE.

8) *AND*, which will be written as && (2 of the &’s) in java, with a general form of:
statement1 && statement2.
Which may be True ONLY when both statements are TRUE.

Open *IntelliJ* and create a new project and name it “Practical03”. You will use this project to complete the checkpoints for this lab.

Task 1: Ticket Code

Start *IntelliJ* and open the project called "Practical03" (download it from FLO).

Open the file `Ticket.java`. Now that you have learned about changing the flow of control, let's write program code that asks users to enter an age and displays the cost of the ticket based on the criteria that was given above. Use both methods to make sure.

Once you are confident that the code works as expected change it so that it displays $6 for people who are 8 years old or younger OR 65 years or older.

Checkpoint 12

Have the program source code and output marked by a demonstrator

Task 2: Nested IF..ELSE

Open the file `NumberEvaluation.java`:

Write a program that prompts the user to input a number. The program should then output the number and message saying whether the number is "Positive, "Negative", or "Zero".

It should further output whether the number is odd or even (zero is counted as even)

Checkpoint 13

Have the program source code and output marked by a demonstrator

Task 3: More IF..ELSE statements

Open the file `CalculateBMI.java`. You need to write a program that accepts user input and then uses these values to calculate the BMI category for that person. BMI is the body mass index of a person and is calculated by taking the weight (in kilograms and dividing it by the square of your height in metres). The following table provides an interpretation of the BMI based on calculated values:

BMI	Interpretation
Below 18.5	Underweight
18.5 - 24.9	Normal
25.0 - 29.9	Overweight
Above 30.0	Obese

Complete the code to output the correct interpretation (must be to one decimal place), i.e. output:

```
Your BMI is 22.4, which means you are in the Normal range.
```

Checkpoint 14

Have the program source code and output marked by a demonstrator

Task 4: Ranges...

A packet of biscuits can hold 12 biscuits, and a box can hold 30 packets of biscuits.

Open the file `Biscuits.java` and write code that prompts the user to enter the total number of biscuits. The program then outputs the number of packets and the number of boxes to ship the biscuits.

Note: Each packet must contain the specified number of biscuits, and each box must contain the specified number of packets. If the last packet of biscuits contains less than the specified number of biscuits, you can discard it and output the number of leftover biscuits. Similarly, if the last box contains less than the specified number of packets, you can discard it and output the number of leftover packets.

For example:

```
Enter the number of biscuits: 360

There are 30 packets of biscuits: 1 box with no leftover packets and no leftover biscuits

Enter the number of biscuits: 458

There are 38 packets of biscuits: 1 box with 8 leftover packets and 2 leftover biscuits

Enter the number of biscuits: 1453

There are 121 packets of biscuits: 4 boxes with 1 leftover packet and 1 leftover buscuit
```

Checkpoint 15

Have the program source code and output marked by a demonstrator

Task 5 (Extension Practice): Reading, Storing and Outputting

1. You are to create a program that asks the user to supply one integer (in the range of 0-10 inclusive).
2. The program should generate a random number within the same range.
3. It should then present a menu to the user and query whether they would like to add, subtract, or multiply the two numbers.
4. Subtraction should subtract the second integer (random generated one) from the first (user supplied).
5. Display an error message if the user supplies an invalid response to the menu prompt.
6. Otherwise, display the results of the arithmetic operation.

Create a program for the above specification (you should consider using the Switch statement or a series of nested If/Else statements).

The following code can be used as a starting point.

```
import java.util.Scanner;

public class simpleCalculator {

   public static void main(String[] args) {
      Scanner scan = new Scanner(System.in);
      int imputNumber;
      int randomNumber;

      int menuNumber;
   // this can only be 1 - addition, 2 - subtraction,
   // or 3 - multiplication

   // 1. Get the input number
   // 2. generate a random number, use Math.random()
   // 3. get the calculation type
   // 4. if the number is not valid output a message
   // 5. else perform the calculation
   // 6. display the results

   }
}
```