ISD Week 1 Lab

A. Writing and running the Hello World program:

1. Click on the Start button (bottom left of the screen). Type `BlueJ` in the search box. Click on `BlueJ` to start executing the Java Integrated Development Environment that we'll be using on this course.
2. Click on `Project` (top left-hand corner) followed by `New Project.` In the `New Project` panel choose a place to save your project (it should default to your home directory) and then add `\Hello` to the end of the folder name. Then press the `Create` button.
3. You should now see a new project window before you with the title `BlueJ: Hello`.
4. Now press the `New Class` button on the side bar to the left of the project window and type `Hello` in the `Class Name` field. Then press the Ok button.
5. Double click the orange box labelled `Hello` and a new editor window should open.
6. Using the Editor in the new window, delete all the code present and type in the following:

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

7. Click on the `Compile` button above the area where you just typed in the code. BlueJ will now save your code to disk and try to compile it. If the code successfully compiles then you should see the message `Class compiled – no syntax errors` in the footer to the editor window.
8. Leave the editor window open and go back to the window entitled `BlueJ: Hello`. Right click your mouse on the orange box labelled `Hello` and select `void main(String[] args)` from the popup window. Another popup should now appear – just press the Ok button. The output to your program should now appear in a new window.

Compile-time errors: What happens to your program if you

- misspell a word e.g. Sstem rather than System
- forget to capitalise a word e.g. system rather than System
- omit a word e.g. System.println rather than System.out.println
- forget the semicolon after ("Hello World!")
- have unmatched curly braces e.g. remove one and see what happens
- omit the quotation marks from the string "Hello World!"

Try making these errors and see what happens.

Run-time errors: What happens if you

- misspell the output e.g. "Hello Word!"
- divide by zero e.g. System.out.println(1/0);

- forget to output e.g. remove the line starting System.out.println

Try making some of these errors and see what happens.

B. Modifying the Hello World program:

1. Modify your program so that the words “Hello” and “World” are printed on two lines.
2. Modify your program so that it prints out the sum of 1+2
3. Modify your program so that it prints out the sum of the first ten positive integers, 1+2+ ... + 10
4. Modify your program so that it prints out the product of the first ten positive integers, 1 x 2 x ... x 10. Use the symbol * to indicate multiplication in Java.

C. Writing some more programs (start a new Project for each one, with a meaningful name).

1. Write a program that prints the balance of an account after the first, second and third year. The account has an initial balance of £1,000 and earns 5 per cent interest per year. Hint: for the first year you need to multiply 1000 by 1.05, for the second year by 1.05 * 1.05 etc.

2. Write a program that prompts the user to enter a number (not necessarily an integer), and outputs the square, cube and fourth power of the number. Use the Math.pow method to compute these three outputs. Test your program with different inputs.

3. Write a program that prompts the user to enter two integers and then prints out their sum, their difference, their product, their average, the larger of the two, and the smaller of the two.

   Hint: to find the larger and the smaller of the two integers, use the `min` and `max` methods declared in the `Math` class of the java.lang package. The classes and methods of the Java library are listed in the Java API (Application Programming Interface) documentation at

   http://download.oracle.com/javase/7/docs/api/

   Open a browser window and go to this site. To find out about a class (in our case `Math`), click on its name in the All Classes list in the left-hand column. You can then find out the methods the class supports – scroll down the main window. To find out about a particular method, click on its name. Note that the methods `min` and `max` are “overloaded”, in the sense that there are several versions of them, for different parameter types. You will be passing integer parameters to them in your program.

4. Write a program that prompts the user for the lengths of the sides of a rectangle (not necessarily whole numbers, so use `double`), and outputs the area and the perimeter of the rectangle.

5. The following pseudocode describes how to compute the price of a meal from the total cost, the additional VAT of 17.5% of the total cost, and the additional gratuity of 10% of the total cost:

   read in the total cost of the meal
   vat = 17.5% * total cost
   gratuity = 10% * total cost
   meal price = total cost + vat + gratuity
   print out the price of the meal

   Translate this pseudocode into a Java program.

6. Write a program that tells a cashier what change to give. The program has two inputs, the amount due, in pence, and the amount received, also in pence. The program should print out the number of £1, 50p, 20p, 10p, 5p, 2p and 1p coins that the customer should receive.

   Hint : use integer division (/) and modulus (%).

   Test your program with positive integers as input. What happens if you enter one or both numbers as negative integers? We will see next week how to constrain and validate user input.

7. Write a program that reads a number between 1,000 and 999,999 input by the user, where the user enters a comma in the input (so you need to read the input as a String). Your program should then print out the number without a comma. Hint: Find the length of the string input. Suppose it contains N characters. Then find the substring consisting of the first N-4 characters and the substring consisting of the last three characters, and print these out.