THE UNIVERSITY OF SYDNEY

INFO1113/COMP9003 Week 1 Tutorial

Compiling, Inputs, Variables and If Statements

Introduction

Welcome to OOP! Introduce yourself to the rest of the class, what degree you are in, why you are interested in programming and what you did over the break.

Each tutorial will contain programming exercises and discussion questions. Use the discussion time to talk to other students in the class and make sure you understand this week's concepts.

Edstem

We use EdStem for questions, tasks and assignments. Please get familiar with Edstem as it will be used heavily through out the semester and is the place where announcements are made. If you cannot login, please notify your tutor so they can address the issue.

Terminal and the File System

Before we learn any Java code we will get familiar with the terminal and unix command line environment. Some useful commands to learn:

cd	Change directory
mkdir	Make directories
ls	List directory contents
rmdir	Remove directory
rm	Remove
mv	Move (also used to rename files)
cp	Copy
pwd	Print Working Directory
diff	display the differences in the files by comparing the files line by line

Here are some directory symbols to help you traverse the file system using the command line:

..	Parent Directory	cd ..
.	Current Directory	./program
-	Previous directory	cd -
~	Home directory	cd
/	Root directory	cd /

Question 1: Creating Your OOP Directory

You can do this in either the Ed workspace or locally on your own computer.

Create a new directory

```
> mkdir OOP
```

Navigate to the directory

```
> cd OOP
```

Create a simple `.java` file.

```
> touch HelloWorld.java
```

Question 2: Java Version

Find out what version of Java is installed on your current machine. Do this by typing `java -version` and `javac -version` into your terminal.

Question 3: Hello World!

Now let's write a simple Hello World program!

Write the following program in Ed or in a text editor of your choice (Gedit, Atom, Vim, Emacs, or whatever you prefer).

```
public class HelloWorld {

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

}
```

You *must* name your file `"HelloWorld.java"` (same as the class name). Save it in your directory that you have created. You are now able to compile it using the `javac` command.

```
javac HelloWorld.java
```

You may notice that a `.class` file has been created. This file contains the bytecode that the `jvm` can execute.

Run the program by entering

```
java HelloWorld
```

Your program should output:

```
Hello, World!
```

Discussion

Let's go through a couple of the key words in the code above. You'll be using them often as you continue with Java. What do each of these keywords mean?

- `public`
- `class`
- `static`
- `void`
- `main`

Input and Output

Input and output differ in Java in comparison to other languages. Java has a large `Reader` and `Writer` library that can be used for a variety of different use cases.

Focusing on just standard input and output, you should have observed from the previous question the method used to output characters to the terminal.

Input is a little different. We will have to wrap a `System.in` object with a Scanner object to read standard input data into our program.

```
Scanner scan = new Scanner(System.in);

String line = scan.nextLine(); //Reads the next line
int number = scan.nextInt();
```

Discussion

Refer to the Java documentation of Scanner and figure out the return type of the following methods. It's important to always know the return type of the method that you called as Java is a strongly typed language.

- `hasNext()`
- `next()`
- `useDelimiter()`

Make a list of the variable types you've encountered in Java already.

What is the difference between a primitive and reference type? Can you classify the types you've listed into the two groups?

Question 4: Meet and greet!

You are tasked with writing a program `MeetAndGreet.java` that will ask for the user's name and greet them with `Hello <name>!`

```
Hi, What's your name? Trent
Hello Trent!
```

Testing your program with .in and .out files

Create a `.in` file and write the input like `Trent`

```
> touch test1.in
```

Create a `.out` file and write the expected output like `Hello Trent!`

```
> touch test1.out
```

Compile and run the file.

```
> javac MeetAndGreet.java
> java MeetAndGreet < test1.in
```

You can manually check the program output or automatically check using `diff` command

```
> diff test1.out <(java MeetAndGreet < test1.in)
```

For more information, please watch the supplementary lecture 1 available in Canvas under Recorded Lectures.

`boolean` type and `if` statements

Like other languages, Java allows code to branch through if-else statements. This is expressed with the following:

```
if (expression)
```

The `expression` must be a boolean expression that evaluates to `true` or `false`. Unlike other languages, Java is strict in the type that is evaluated within the if statement and the type must be `boolean`.

```
int x = 5;
boolean exp = x < 6;
if(exp) {
        System.out.println("x is less than 6");
}
```

Or

```
int x = 5;
if(x < 6) {
        System.out.println("x is less than 6");
}
```

Question 5: Grade Program

Write a program that will output the grade a student will be awarded based on the mark given as a command line argument.

Your program should output:

- if the grade is >= 85, the student will be awarded a "High Distinction"
- if the grade is >= 75 and < 85, the student will be awarded a "Distinction"
- if the grade is >= 65 and < 75, the student will be awarded a "Credit"
- if the grade is >= 50 and < 65, the student will be awarded a "Pass"
- if the grade is >= 0 and < 50, the student will be awarded a "Fail"

Question 6: Die roll

You are to write a program that simulates a die roll. Your program should output the roll. You can either `import java.util.Random` or use `Math.random()` for this task.

Remember: A common die has 6 sides and is numbered 1 to 6.

```
> java DieRoll
5
```

Question 7: Sort 3 integers

Write a program that will take in 3 integers as command line arguments and rank them from largest to smallest. Using only if statements, output the integers in the correct order.

```
public class Sort3Integers {

        public static void main(String[] args) {
                int x = Integer.parseInt(args[0]);
                int y = Integer.parseInt(args[1]);
                int z = Integer.parseInt(args[2]);

                //Your code here
        }
}
```

Question 8: Assessed Task: Online Task 0 (Practice)

Remember you are required to complete a Online Task within the due date. Go to EdStem and click on Lessons to find out the task and the due date. This will be a practice task to familiarise yourself with the assessment submission procedure. Note that you are allowed to submit multiple times but only the last one will be marked.