159.261 Lab 1
Rapid Introduction to Java

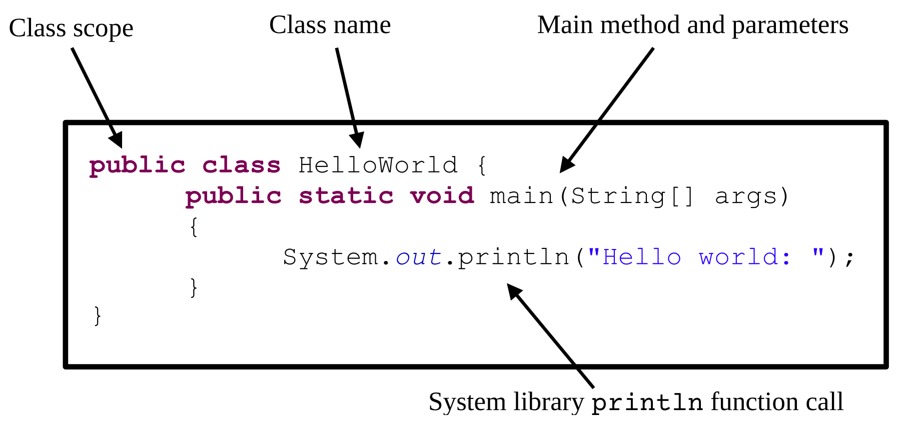
The best way to learn Java (for those of you who don't know Java) is to actually grab a compiler and start coding. There is a wealth of information about Java and Java Programming available on the Internet, as specified in the lecture notes.

There are 9 questions in this lab. Please attempt all questions.

1) Hello World

Read through the getting started trail on the Oracle website:
http://docs.oracle.com/javase/tutorial/java/

Make sure you pay attention to the section on the Java platform, so you understand how Java programs are executed using a JVM and how Java programs are compiled to bytecode. After finishing reading through these sections develop and run your own version of the HelloWorld program.


Class scope
Class name
Main method and parameters
System library println function call

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

Compile and run your program. Make sure you are able to identify the class definition, class name, class scope, main method and method scope.

2) Command Line Arguments

Modify the hello world exercise so that your program is able to accept command line parameters which specify the user's name.
For example, if the HelloWorld program were run with these parameters:

```
java HelloWorld Jack
```

The program would output the following message.

```
Hello World: Jack
```

3) Understanding Code

One important aspect of learning a new programming language is learning how to read code and predict or understand what it is going to do (you spend far more time reading code than writing it).

When a piece of code doesn't behave the way you expect, it's a chance to correct a misunderstanding about the language.

What will the output of the following code be if `x=6` and `y=9`?

What about if `x=5` and `y=2`?

```
if (x > y)
{
   int t = x;
   y = x;
   x = t;
   System.out.println("x = " + x + " and y = " + y);
}
else
{
   System.out.println("Have a nice day!");
}
```

4) Modifying Code

There's more than one way to write a program, experimenting with different features helps you get to grips with a language. Don't just learn one way of doing something and always stick to it.

What is the output of this program if `6` is given as a parameter?

```
java PowersOfTwo 6
```

Replace the '`while`' loop with a '`for`' loop and run the program again.

Make sure it produces the same output.

```
public class PowersOfTwo {
   public static void main(String[] args) {
      // last power of two to print
      int N = Integer.parseInt(args[0]);
      int i = 0;

      // loop control counter
      int v = 1;

      // current power of two
      while (i <= N) {

         System.out.println(i + " " + v);
         i = i + 1;
         v = 2 * v;
      }
   }
}
```

5) More Practice

Compile and run the following code using 3 as a parameter (`java Cubes 3`).
What is the output?

```
public class Cubes {
  public static int cube(int i) {
    i = i * i * i;
    return i;
  }

  public static void main(String[] args) {
    int N = Integer.parseInt(args[0]);
    for (int i = 1; i <= N; i++)
      System.out.println(i + " " + cube(i));
  }
}
```

6) Working with classes

Define a new class: `TestCubes`

It should have a `main` method in which it creates objects of type `Cubes`.

Remove the main method from `Cubes`.

Compile the classes run `TestCubes` with `3` as a parameter

```
java TestCubes 3
```

It should produce the same output as exercise (5).

7) Vectors

Given two arrays `x[]` and `y[]` of length `N`, their dot product is the sum of the products of their corresponding components.

Study the following code and fill out the ? marks in the given table.

```
double[] x = { 0.3, 0.6, 0.1 };
double[] y = { 0.5, 0.1, 0.4 };
int N = x.length;
double sum = 0.0;

for (int i = 0; i < N; i++) {
    sum = sum + x[i]*y[i];
}
```

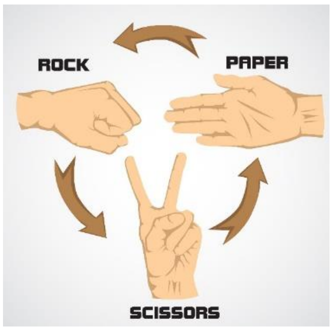
i	Sum
0	?
1	?
2	?

8) Matrices

Given two N-by-N matrices `a` and `b`, define `c`, the N-by-N matrix where `c[i][j]` is the sum `a[i][j] + b[i][j]`. You may use nested `for` loops.

9) A Simple Game

In this exercise you will develop a simple text-based Rock, Paper, Scissors game that you can play against the computer.



Your program should carry out the following steps:

1. Player Turn - Get the input choice (Rock, Paper, Scissors) from the player. It must validate the input to check if the player has given proper input.
2. A.I Turn - Generate a random choice (Rock, Paper, or Scissor)
3. Check for the winner or a draw. Increment the points for the winner. No points are awarded if it's a draw.
4. Continue the game till one of the players has scored 3 points.
5. Print the winner and total points.

Below is the Table which displays the winning results between two choices

	Rock	Paper	Scissor
Rock	Draw	Paper	Rock
Paper	Paper	Draw	Scissor
Scissor	Rock	Scissor	Draw

For example:
Player turn (Rock) vs A.I. turn (Rock) results in a draw
Player turn (Rock) vs A.I. turn (Paper) results in a Paper win (A.I. win).
Player turn (Rock) vs A.I. turn (Scissor) results in a Rock win (Player win) and so on.

Bonus:
Try to think how you can use a multidimensional array (2d array) instead of nested if-else statements or switch cases when deciding the winner.