Informatics 1: Object Oriented Programming

Tutorial 01

Week 2: 21/01 - 25/01

Volker Seeker (volker.seeker@ed.ac.uk)
Naums Mogers (naums.mogers@ed.ac.uk)

1 Introduction

The goal of this first tutorial is for you to familiarise yourself with your development environment, i.e. ECLIPSE and JUNIT tests.

ECLIPSE is a widely used integrated development environment (IDE) and well suited for working with JAVA. You might already have a different preferred IDE such as NETBEANS or INTELLIJ. This is fine, you can keep using it if you like. However, keep in mind that you will only have ECLIPSE available during the exam. You should make sure that you are comfortable using it. The mock exam is a good place for a final test. In this course we are using ECLIPSE PHOTON 4.8.0.

The same is true if you do not use an IDE at all but prefer an editor and the command line. You will definitely have access to the command line during the exam and it is likely that your editor is available. To be certain, you should double check with Computing Support and during the mock exam.

JUNIT is a framework for unit testing your code. It will test the semantic correctness by running single units of it against various test cases. In this course, you do not need to learn how to write your own unit tests. However, you need to know how to execute JUNIT tests and interpret the results. Also, you need to be able to configure Eclipse or the command line in a way that allows you to make use of JUNIT. For this course we are using JUNIT version 4.

Each lab exercise and future tutorial exercises will come with their own unit tests. The lab exercise tests provide large semantic coverage and you can see if you solved the exercise correctly or not. The tutorials, like the exam, come with only basic tests which check your function headers to make sure you are going in the right direction. Please ask for feedback from your tutor regarding your tutorial solutions.

Both ECLIPSE and JUNIT are already setup and installed on each DICE machine.

2 Homework

Before the tutorial, please work through the ECLIPSE and JUNIT introduction you can find on the lab exercise pages:

> http://www.inf.ed.ac.uk/teaching/courses/inf1/op/2019/labs/beginjunit.html

You can do this on any DICE machine or if you prefer on your own computer. If you would like to use your own computer, it is best to use the same ECLIPSE and JUNIT versions which are used on DICE.

Optional For a deeper understanding of automatic testing, you can work through this additional tutorial which describes the inner workings of a unit test and demonstrates how to execute them using the command line:

> http://www.inf.ed.ac.uk/teaching/courses/inf1/op/2019/labs/usingjunit.html

3 Tutorial Exercises

To make the entry easier, this week's exercises can be done during the tutorial with the help of your tutor. Please note, that some DICE machines are available during the tutorial but likely not for everyone. Please bring your own laptop if you can.

Future tutorial exercises should always be attempted at home first before you go to your tutorial. This is important! Simply reading the solution does nearly nothing to further your own programming skills. Even worse, knowing the solution up front will spoil the opportunity for you to practice your programming skills with the given exercise afterwards.

Use ECLIPSE to find the bugs in the following four programs. Consider using the following techniques:

- interpret errors being pointed out by ECLIPSE before compilation (Quickfix option)
- interpret compiler and runtime errors
- use primitive print statements throughout the code to figure out what is wrong
- comment in bits of code one by one to find the problem
- use the ECLIPSE debugger
  - set breakpoints or conditional breakpoints
  - step through code
  - put down variables or expressions to observe the content

You can find the JAVA files to all exercises in the schedule table of the course content page:

```
https://www.learn.ed.ac.uk/webapps/blackboard/content/
listContent.jsp?course_id=_63615_1&content_id=_3012072_1&mode=
reset
```

Download them and import them into your ECLIPSE JAVA project like this:

- Right-click `src`
- Select `import`
- Select `General - File system` and click `Next`
- Navigate to your download directory which is the directory to import from
- Once you can see the files listed you want, click `OK`
- Select the `.java` files and click `Finish`

Task 1 - Errors 01

◁ Task

```
class Error01 {

        public static void main(String[] args) {
                int m;
                int n = 5

                System.out.println("Starting with " + m);

                while(n > 0) {
                        System.out.println(n);
                        n--;
                }

                System.out.printn("No errors found");
        }
}
```

Task 2 - Errors 02

◁ Task

```
public class Error02 {

        public static void main(String[] args) {
                String[] names = {"Peter", "Paul", "Petra", "Pit", "Peggy"};

                for (int i = 0; i <= names.length; i++) {
                        System.out.println(names[i]);
                }
        }
}
```

Task 3 - Errors 03

◁ Task

```
public class Error03 {

        public static void main(String[] args) {
                int n = Integer.parseInt(args[0]);
                System.out.println("Printing the first " + n + " powers of 2.");

                int i = 0;
                int val = 1;

                while (i <= n) {
                        System.out.println(i + " " + val);
                        val = 2 * val;
                }
        }
}
```

Task 4 - Errors 04

◁ Task

```
public class Error04 {

        public static void main(String[] args) {
                int n = Integer.parseInt(args[0]);
                System.out.println("Printing the first " + n + " powers of 2.");

                int i = 0;
                int val = 1;

                while (i <= n)
                        System.out.println(i + " " + val);
                        i = i + 1;
                        val = 2 * val;
        }
}
```