

# INTRODUCTION TO PROGRAMMING - CONCEPTS AND TOOLS

## ASSIGNMENT 1

### GENERAL INFORMATION

This assignment is made public on Friday, August 30, 2002. The assignment is due on

Friday, September 6, 1 PM.

Hand in your assignment to the teaching assistant running your lab session.

The first page of your (written) assignment has to contain at least the following information:

- the course name (Introduction to Programming - Concepts and Tools)
- your name and your student number
- name and student number of the fellow student if you submit in pairs
- assignment number

*Please staple your assignment!*

You will get back the graded assignment one week after submission deadline.

### QUESTIONS

#### 1. *Lexical Elements*

Download the program `Literals.java` from the course homepage (follow the link to labs-assignments). The program is almost identical to the program `MakeChange.java` of Chapter 2 of the textbook except for the following changes: (i) the file name is changed, (ii) the name of the main class is changed (since the class name has to be the same as the file name); and (iii) the file contains extra empty lines.

Print the file. Mark either in color, or with different types of underlining, the different lexical components of this Java program. Next to the lexical component you will find the colour (or type of underlining) that you should use:

- comments; (yellow)
- keywords; (blue)
- identifiers; (black)
- literals; (green)
- operators and punctuation; (red).

#### 2. *Identifiers*

Which of the following are allowed identifiers (and could, for example, be used as variable or function names)? If the identifier is not allowed please give a short explanation why this is the case.

- |                                 |                              |
|---------------------------------|------------------------------|
| (a) <code>firstComponent</code> | (f) <code>true</code>        |
| (b) <code>1stComponent</code>   | (g) <code>True</code>        |
| (c) <code>ComponentOne</code>   | (h) <code>class room</code>  |
| (d) <code>componentone</code>   | (i) <code>classRoom</code>   |
| (e) <code>Component_One</code>  | (j) <code>(classroom)</code> |
|                                 | (k) <code>\$Local</code>     |

### 3. *Types and variables*

In a program you see the following assignments which involve variables, literals, and methods like `Math.sqrt()`. What can you say about the type of all involved variables?

- (a) `a = '7';`
- (b) `b = "Hello Student!";`
- (c) `c = cc + 3;`
- (d) `d = Math.sqrt(dd + 4)`
- (e) `e = 'e' + 4;`
- (f) `f = (int)(ff + 0.5);`

### 4. *Error messages*

We haven't talked much about error messages, but you will see more of them than you like. Type in the following program `SyntaxError.java` or download it from the course homepage.

```
Class SyntaxError{
    public static void main(String[] args) {

        System.out.println(hello world);

    }
}
```

The program contains three syntax errors and when you try to compile it produces the following error message:

```
>javac SyntaxError.java
SyntaxError.java:1: 'class' or 'interface' expected
Class SyntaxError{
^
1 error
>
```

(The message starts with the fact that the compiler found a syntax error in line 1, and gives then some clues as to what the problem might be. Then the problematic line is printed and a little marker is placed.)

Thus, only the first error is revealed. Correct the error, save the file, and recompile the program. This time you will get three error messages, even though there are only two errors left!

```
>javac SyntaxError.java
SyntaxError.java:4: ')' expected
System.out.println(hello world);
                        ^

SyntaxError.java:6: '}' expected
}
^

SyntaxError.java:4: cannot resolve symbol
symbol : variable hello
location: class SyntaxError
System.out.println(hello world);
                        ^

3 errors
>
```

For each of these three error messages describe in your own words the following:

- What do you think the problem is?

- Does the compiler make a suggestion how to correct the error? Is the suggestion 'correct', i.e., does the compiler point to the real problem?

Hint: The first of the three messages is confusing.

## 5. Programming

You are supposed to write a Java program that allows the user to input the radius of a sphere and outputs its volume. Recall that the volume of a sphere is calculated by the formula

$$V = \frac{4 \times \pi \times r^3}{3}$$

where  $r^3$  is  $r \times r \times r$  ( $r$  to the power of 3), and  $\pi$  is the mathematical constant 3.14159... To do so modify the program `SimpleInput.java` from Chapter 2 of the textbook, which performed a similar task: asking the user for input, doing some calculation, and then output a result on the screen.

- Copy the file `SimpleInput.java` from the course homepage. The program is from Chapter 2 of the textbook.
- Change the name of the file to `Volume.java` (either use the Linux command

```
mv SimpleInput.java Volume.java
```

which renames the file, or open the file `SimpleInput.java` with `emacs` and then use the command `Save Buffer As ...` from the pull-down menu `Files`).

- Open the file using `emacs` (if you haven't done so already).
- Modify the comments at the beginning so that they contain at least your name, the date when you write your program, and the purpose of the program.
- Change the name of the main class to `Volume`. (Why?)
- Change the body of the `main` class so that you ask the user to input the radius of the sphere, input the sphere, do the calculation, and output the result. Make sure that the user of your program will see some informative text on the screen when running the program. The mathematical constant  $\pi$  can be accessed through `Math.PI`. It is of type `double`.
- Compile your program. If you get error messages, read them carefully and try to correct your code. Most likely you will make some typing errors, or omit braces or semi-colons.
- Run your program.
- Print your final program once you are satisfied with its performance. Submit that paper copy.

## 6. Assignments

Suppose you are sitting at the computer and have no book or other reference material at hand. A fellow student asks you whether the following assignment

```
double min = 0; max = 12;
```

is correct Java syntax declaring two variables `min` and `max` of type `double` setting their values to 0 and 12 respectively.

Write a short program that will test whether (i) the above syntax is correct, and (ii) whether the variables are really set to those values. (If the above syntax is correct the program will compile, and then you have to check whether the variables really contain those values.)

Check whether the syntax above is correct by running your program. If the syntax is incorrect you will receive an error message. In that case report the error message. Can you understand the message? Does it help you finding the syntax error? What correction do you have to make to get your program running (and thus the above syntax declaration correct)?

Submit your test program with the error message that is produced, and give the correction necessary to compile the program without error.

Hint: Again the easiest is to take one of the small programs you have seen already and modify them according to your needs.