

## **JAVA PROGRAMMING – EXERCISE 3**

In this lab, we'll be accepting user inputs, working with mathematical operations, if-statements and switch statements.

### **COURSE MARKER**

1) Start Course Marker by selecting

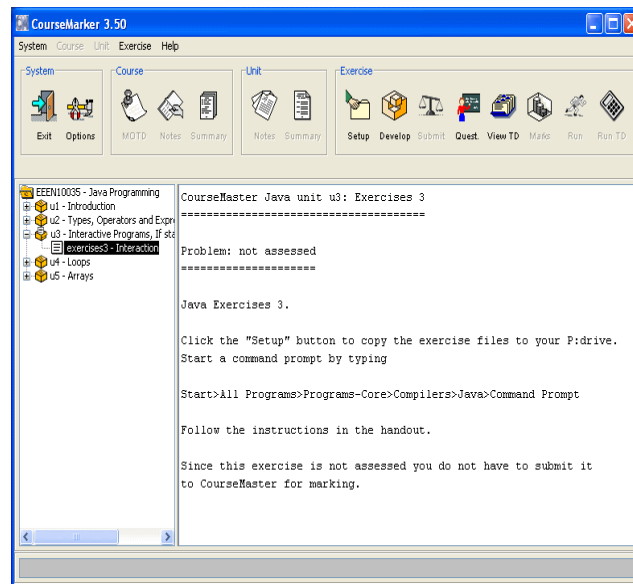
2) **Start>All Programs> CourseMarker >CourseMarker**

*Note: if CourseMarker is not available you can download the files needed from Blackboard, under Exercise Files. Right-click on each one and “Save link as...”, or “Save target as...” and save each file to your P: drive.*

After a few moments you will see the login window -the UserName is your usual username (mch...), and the password is your student ID number.

3) Once you're in, access the relevant assignment by selecting:

**EEEN10035 – Java Programming>u3 – Interactive Programs**



4) Click the Setup button on top, and then click OK.

Clicking the Setup button copies the files needed for this exercise to the folder P:>mchXXYY> EEEN10035u3exercises3 (where mchXXYY is replaced with your username).

The files copied are:

MathExercise4.java, MathExercise5.java, ScannerExample.java, PrintAverage.java, PrintSmallest.java, PrintLargest.java, and Switch.java. These are the java files you'll be working on this time.

- 5) That's all we need from Course Marker, so now you can close it.
- 6) Everything we do from here on out will involve *notepad* and *command prompt*.
- 7) First off, open command prompt via, **start >All Programs** and scroll down until you see the folder **JDK** –click it and then click **cmd.exe**

In the resulting window type

**P:**    <press enter>

and then type

**cd mchXXYY\EEEN10035u3exercises3** <press enter>

(We're just navigating to the folder where course marker copied all the necessary java files.)

- 8) Now, type the command **dir** and press enter.  
You should see the following files in the folder: MathExercise4.java, MathExercise5.java, ScannerExample.java, PrintAverage.java, PrintSmallest.java, PrintLargest.java, and Switch.java.
- 9) Before we move on ahead, we need to learn how to interact with the user (i.e. accept some form of user input) while running a java program.

## USER INPUTS

A major aspect of any program is to interact with the user.  
Let's look at a really basic program,

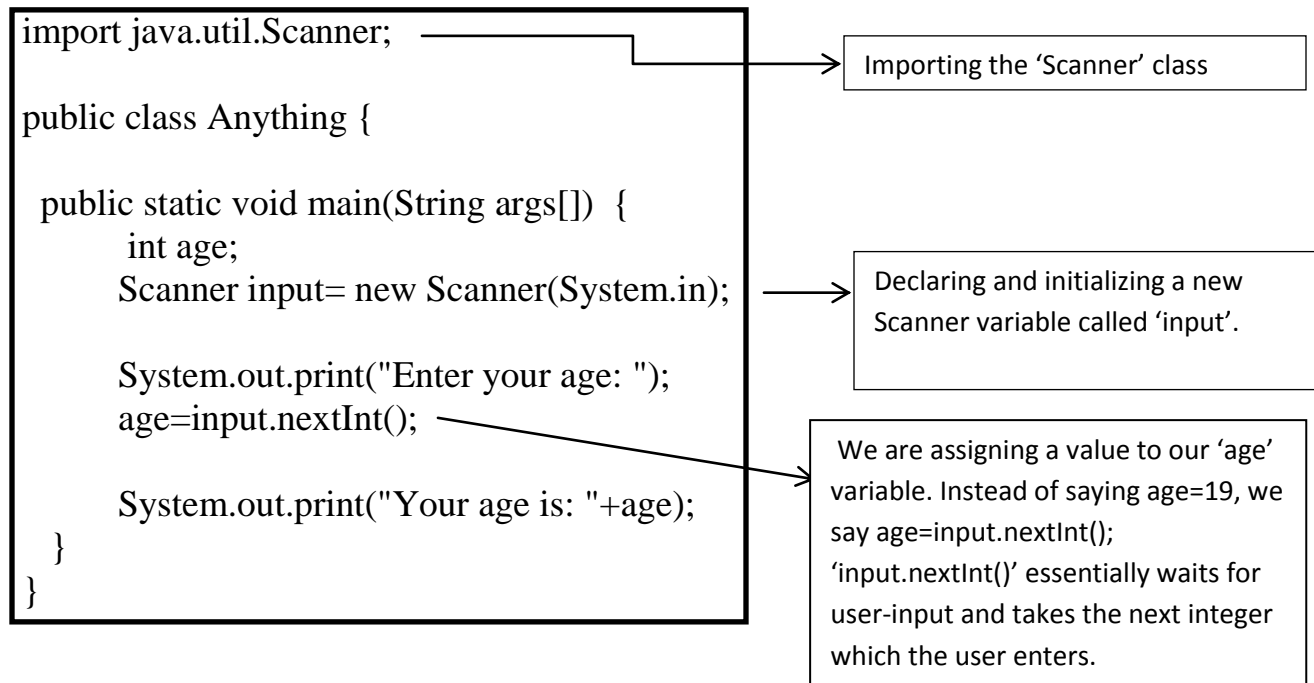
```
public class Anything {  
  
    public static void main(String [ ] args) {  
        int age= 19;  
        System.out.println("Your age is : " + age);  
    }  
}
```

```
}
```

Obviously, the output of this program is based on a pre-defined number (19). Wouldn't it be better if the program could prompt the user to enter his/her age?

We can do that by using an **imported** Java library called 'Scanner'.

Study the modified program below, line by line:



### Video Help?

<http://stream.manchester.ac.uk/Play.aspx?VideoId=9636>

As you noticed, we entered "age=input.nextInt();" in the previous program, because age was an int variable. If 'age' was a double variable, it would have been "age=input.nextDouble();".

You must use the correct Scanner method for the type of input expected: **nextInt()** for integers, **nextDouble()** for doubles, **nextBoolean()** for booleans, etc. If the program is expecting the user to enter an integer and he/she enters a fractional number, the program will terminate with an error.

10) Now, let's put this new found knowledge to the test.

In your command-prompt window, type in **notepad ScannerExample.java** and press enter.

### Task 1:

Complete, compile and run ScannerExample.  
The completed version should look like this;

```
import java.util.Scanner;
public class ScannerExample {
    public static void main(String [ ] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter a number: ");
        double a = input.nextDouble();
        System.out.println("You entered: " + a);
    }
}
```

Run and test the program with a non-number, say, a character such as 'a', or 'b'.  
Observe what happens.

### Task 2:

Now let's program something useful. Open the PrintAverage.java file using notepad. Develop PrintAverage.java so that it prompts the user to enter three numbers (use doubles), and prints out their average.

## MATHEMATICAL FUNCTIONS

11) You already know about the operators + - \* and /.

There is another operator which gives you the remainder between two numbers,  
**the '%' sign** –  
e.g.) 11%5 is equal to 1, because the remainder is 1

12) It is very important to note that there is a **well-defined order of priority**  
between these operators.

Operator	Meaning	Priority
-	negation	high
* / %	multiplication, division, remainder	medium

+ -	addition, subtraction	low
-----	-----------------------	-----

- 13) Expressions are usually evaluated left to right, so that using the precedence (or priority) rules above, an expression such as

**int i = 4 + 3 \* 6 % 4 \* 5;**

is evaluated as:

4 + 18 % 4 \* 5

4 + 2 \* 5

4 + 10 = 14

- 14) Parentheses are used to form groupings, e.g. (4 + 3) \* 6 % 4 \* 5 = 10  
The expressions enclosed within the ( ) are evaluated *first*.

- 15) Expressions that contain a divide-by-0 result in an error, e.g. double t = 3/0;  
-this statement will compile, but results in an error (Java calls it an Exception) when the program is run.

### Task 3:

Add printing statements to MathExercise4.java that displays the results of the following integer expressions (compile and run to determine the result).

*Hint: the expression can be directly placed inside a print statement, for example System.out.print(the mathematical expression)*

**12 \* 3 / 6 \* 8 / 4**

**10 + 3 % 3 + 4**

Try to predict or estimate the results of:

**20.0 - 2 / 6**

**10.3 / 6.2 \* 3.2 + 4.1**

Test your predictions by adding the statements, re-compiling and running the program again.

## INCREMENT AND DECREMENT OPERATORS

- 16) An increment operator is written as '++' and helps to add 1 to a number.

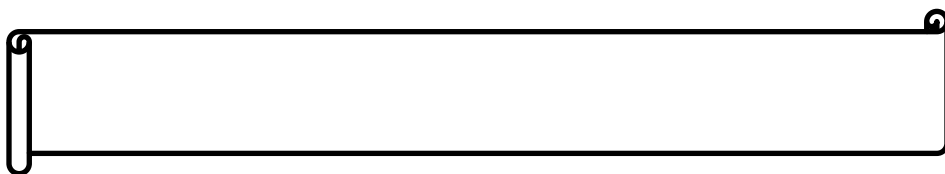
17) A decrement operator is written as '--' and helps to subtract 1 from a number.

Examples:      `int a = 5;`  
                  `a++;`  
                  `System.out.print(a);`      → The output will be 6

`int a=5;`  
`a--;`  
`System.out.print(a);`      → The output will be 4

18) There are two types of increment operators.

Post-Increment	Pre-Increment
<p>Example: <code>int a = 5;</code>           <code>a++;</code></p> <p>As you can see the '++' is written after the variable name, i.e. this is a post-increment operator.</p>	<p>Example: <code>int a = 5;</code>           <code>++a;</code></p> <p>As you can see the '++' is written before the variable name, i.e. this is a pre-increment operator.</p>
<p>The post-increment operator updates the value of the variable, after the statement is executed.</p> <p>Example: <code>int a = 5;</code>           <code>System.out.print(a++);</code></p> <p>The output of this program will be 5. Why? Because the value of 'a' is updated <u>after</u> it has been printed out, i.e. <u>after</u> the print statement has been executed.</p>	<p>The pre-increment operator updates the value of the variable, while the statement is being executed.</p> <p>Example: <code>int a = 5;</code>           <code>System.out.print(++a);</code></p> <p>The output of this program will be 6. Why? Because the value of 'a' is updated <u>before</u> it's printed out, i.e. <u>while</u> the print statement is being executed.</p>



## Video Help?

<http://stream.manchester.ac.uk/Play.aspx?VideoId=9637>

- 19) Similarly, there are post-decrement and pre-decrement operators too, which work in the same way.

### Task 4:

What would be the output given the following statements? Test by adding them to MathExercise4.java, compiling and running:

```
int vara = 3;
int varb = 4;
vara++;
--varb;
System.out.println("vara = " + (++vara) + ", varb = " + (varb--));
```

### Task 5:

The increment and decrement operators have a higher precedence than the other arithmetic operators. Try to predict the output of the following statements:

```
int i = 3, j = 4;
double k = i++/--j * 4.0;
System.out.println("k = " + k);
```

Again test this by adding them to MathExercise4.java, compiling and running.

## MATH CLASS

There is a dedicated math class in java, which can help you deal with logarithms, square roots, powers, etc.

Method	Meaning
<b>Math.PI</b> Example: System.out.print(Math.PI);	Gives you the value of $\pi$
<b>Math.sqrt()</b> Example: System.out.print(Math.sqrt(25)); Output: 5	Gives you the square root of a number

<b>Math.abs()</b> Example: <code>System.out.print(Math.abs(-53));</code> Output: 53	Gives you the magnitude (absolute value) of a number
<b>Math.pow()</b> Example: <code>System.out.print(Math.pow(5,3));</code> Output: 125	Raises the first number to the power of the second number.
<b>Math.log()</b> Example: <code>System.out.print(Math.log(10));</code>	Gives the natural log of the number in brackets.
<b>Math.E</b> Example: <code>System.out.print(Math.E);</code>	Gives you the value of the exponential $e$

Other Examples:

- a) `double radius=5.5;`  
`double circumference = 2*Math.PI*radius ;`
- b) `double numb=3*Math.E;`  
`double r=Math.sqrt(36);`
- c) `double y = Math.abs(-3.2);`  
`int x = Math.abs(-22);`
- d) `System.out.println("The absolute value of -22 = " + Math.abs(-22));`

To see the complete list, go to <http://docs.oracle.com/javase/6/docs/api/>, and search for ‘Math’ under ‘All Classes’.

### Task 6:

Open MathExercise5.java using Notepad.

Using appropriate methods from the Math class, add statements to MathExercise5.java to print out:

- the square root of 6.37
- the absolute value of  $a^2 - b^2$   
*[use your own choice of type and values for a and b]*



- $\sin(\pi)$   
*[Find out the math method from the complete list online]*

### Task 7:

This next step involves the use of **if-statements** that were covered in the recent lecture notes.

#### Video Help?

<http://stream.manchester.ac.uk/Play.aspx?VideoId=9638>

- Develop the file PrintLargest.java that prompts the user to enter two doubles and prints the largest. Compile and test the program.
- Develop the file PrintSmallest.java that prompts the user to enter two integers and prints the smallest. Compile and test the program.

### Task 8:

This next step involves the use of **switch** statements that were also covered in the lecture notes.

Develop the file Switch.java so that it prompts the user for one of the numbers 1, 2, 3, or 4. These numbers must mean “up”, “down”, “left” and “right” respectively. The program should use a switch statement to determine what was entered and print out the meaning, e.g. if the user entered 1 the output “up” should be printed. If one of these 4 numbers is not entered the program should print out the message “unknown action”.

Compile and test the program.

### Task 9:

There are some additional , “advanced”, java exercises, see “Week 3” under Further Exercises on Blackboard. See how many you can complete!

**END**