# Object-Oriented Development (CIS1056-N) Worksheet 01: NetBeans and Primitives

## Before You Start

Before you touch your keyboard, read through this brief to get an overview of the task at hand. You are not expected to complete the entire brief within the allotted two hours, but to make a start and continue outside of the class.

Attempt to complete this set of tasks before your next session. Any issues seek help from your tutors.

## Introduction

The learning aims of this session are to familiarise you with the NetBeans IDE by creating a project from scratch and implementing some basic Java programs.

## Creating your first Java Program

Linux and Java are both case-sensitive (jpr and JPR are different). In the following activities please take particular care over the how you type the names of folders and files.

### 1. Create a Folder for Your Programs (Linux)

* From the desktop open the nethome space folder.
* Create a folder for the module: Select **File -> Create Folder** and enter the name **jpr**.
* Open the **jpr** folder.
* Create a folder for week 1: **Select File -> Create Folder** and enter the name **week01**.

To help manage your work, we suggest that you create a new folder for each week’s activities.

### 2. Create a new Java Project in the NetBeans IDE

1. Start NetBeans by selecting **Applications -> Programming-> Applications -> NetBeans IDE**. If this is the first time you have started NetBeans, the following screen will display:

Graphical user interface, application

Description automatically generated

1. Select File > New Project. The New Project Dialogue will appear:

Graphical user interface, text, application

Description automatically generated

1. Ensure **Java** in selected in Categories and **Java Application** in Projects and click **Next**. The New Java Application Dialogue will appear:

Graphical user interface, text, application, email

Description automatically generated

1. Change the Name and Location properties to:

|  |  |
| --- | --- |
| Project Name: | **Welcome** |
| Project Location: | **/nethome/a1234567/jpr/week01** (*a1234567* will be your user id) |
| The Project Folder and the Main Class name will automatically update. | |
|  | |

1. Ensure Create Main Class is ticked.

Graphical user interface, text, application, email

Description automatically generated

1. Click Finish to create the project files. The following will be displayed.

Graphical user interface, text, application

Description automatically generated

Navigation Panel Source code goes here

Project Panel

### 3. Update the code to match the following

/\*

 \* First Java Program

 \*

 \*/

package welcome;

/\*\*

 \*

 \* @author School of Computing, Engineering & Digital Technologies

 \*/

public class Welcome {

    /\*\*

     \* @param args the command line arguments

     \*/

    public static void main(String[] args) {

        System.out.println("Welcome to Java Programming");

    }

}

### 4. Execute the program

Select **Run -> Run Project (Welcome)**. The Output Window will appear at the bottom of the main window:

Graphical user interface, text, application

Description automatically generated

Output Window

## Java Primitives

Here, you will write Java code creating and manipulating primitives. Use the following online resource to help with your primitive choices:

<https://www.baeldung.com/java-primitives>

### 5. TestPrimitives Project

Create a new NetBeans Project called **TestPrimitives**. Inside the main method of this class, declare (but do not *initialise*) the following primitive variables:

* numberOfSweets that stores a whole number value.
* isPregnant that stores a true/false value.
* itemPrice that stores a value in pounds and pence.
* studentGrade that stores the grade received by a student (e.g. A, B, F).

Run the **TestPrimitives** Project and fix any compiler errors.

### 6. Initialise primitives

Having declared the primitive variables in (5) you will now initialise them. Underneath the code you have already written, initialise the variables with the following values:

* numberOfSweets should be initialised to 41.
* isPregnant should be set to false.
* itemPrice should be initialised to 17.99.
* studentGrade should be initialised to a C.

Run **TestPrimitives** and fix any compiler errors.

### 7. Check the values

Now it is time to check the values stored inside your variables.

1. Underneath the code you have written so far, write the following code to print out the value stored inside the numberOfSweets variable:

System.out.println(numberOfSweets);

Run **TestPrimitives** and fix any compiler errors. Your application should print '41' to the output window.

1. Add three more lines of code to TestPrimitives.java which print out the values stored in the remaining variables - isPregnant, itemPrice, and studentGrade.

### 8. Extend the TestPrimitives program

Extend the **TestPrimitives** program, on one line, declare AND initialise a variable called weightInKilos that has a value of 77. Print this value to output window as shown in question 7.

Declare three more variables (a boolean, a char, and a double) of your own choosing using this one-line technique. You will have to come up with the name and values yourself.

### 9. Assign the result of an expression

In Java, it is possible to assign the result of a calculation to a primitive variable. For example, you could write:

int x = 5 + 7;  // i.e. x = 12

Create a new NetBeans Project called **TestArithmetic**. Inside the main method, implement the following:

// Write code declaring three variables of type int, called a, b and c

// Initialise these variables to 3, 5 and 10 respectively.

// Write code here assigning the sum of a and c to a new int

// variable called d

// Print out the value of d. It should be 13

// Write code here assigning the product of a and b to a new

// int variable called e

// Print out the value of e. It should be 15.

// Write code here assigning the remainder when c is divided by a to a

// new int variable called f

// Print out the value of f. It should be 1.

// Divide c by a and assign the result to a new int variable called g.

// Note that the result is technically incorrect (i.e. no decimal places).

// Repeat the operation above, assigning the result to a double

// variable h.  Note the result is still technically incorrect

// (i.e. there is a decimal place, but it reads 0).

// How can you get the correct answer (3.33333333)?

// Declare an int variable called Celsius. Initialise it to 30.

// Now print out the Fahrenheit equivalent of this value.

// To convert Celsius temperatures to Fahrenheit you

// multiply the temperature by 9/5 and add 32. If you are not getting

// the right answer, review your notes on operator precedence.

// Declare an int variable called Fahrenheit. Initialise it to 0.

// Now print out the Celsius equivalent of this this value.

### 10. What is the output produced by the following lines of code?

        int n = (int)3.9;

        System.out.println("n == " + n);

Create a new Java project called **Numbers** in NetBeans and add the above code to the main method to check the output is what you expected.

### 11. What is the output produced by the following lines of code?

        double number = (5/7) \* 7;

        System.out.println("5/7 \* 7 is equal to " + number);

Add the above code to the **Numbers** project to check the output is what you expected.

### 12. Assigning values of other variables

Underneath the code you have written so far in **TestPrimitives**, declare a primitive of type int with the identifier first. Then declare another primitive of type int with the identifier second. Assign first the value 17. Now assign second the value stored in first. Print out the value of second.

Experiment with other assignments where the value assigned to a variable is not a literal value (e.g. 17, 3.2) but the value stored in another variable. For example:

        char badGrade = 'D';

        char studentGrade = badGrade;

### 13. How the Java compiler handles errors

To demonstrate how the Java compiler handles errors, you will now try to assign the wrong sort of value to a primitive variable in the **TestPrimitives** program.

You will assign a value with a floating point to the numberOfSweets variable. What type of error does this generate? How can you force this assignment? When you have forced the assignment, what is the value stored in numberOfSweets?

Try assigning other incorrect values to your variables (e.g. a value of true to studentGrade). What happens?

### 14. Formula to Java

Convert each of the following mathematical formulas to a Java expression:

## Document History

Revision 0 (19-Sep-22): This is the initial version of the 2022/23 exercise.