

Lab 01 – Integrated Development Environment (IDE)

Objectives:

- Learn to use Netbeans IDE to create, debug, compile, and execute a project
 - Perform some basic programming tasks with Java
1. Here is a walk-through of the basic functions in **Netbeans**. Follow the instructions from your tutors and finish it carefully.

A. Create an Empty Project

- i. From the menu bar, choose: File > New Project
- ii. Select “Java” in the Categories and “Java Application” in the Projects
- iii. Enter the project name and location and uncheck the “Create Main Class” option
- iv. Click “Finish” to create the project

B. Add a New Class and a New Package

- i. Right click the project and choose: New > Java Class
- ii. Enter the class name and package name (if no package name is provided, you will see the new class is created in a default package)
- iii. Using the same procedure, add a new package to your project: New > Java Package
- iv. Try to drag and drop your class around different packages. You will be prompted for “Refactoring”. Click to confirm
- v. After refactoring, see what change is applied to your source code
- vi. Also, look at your project folder, see how the source code file (*.java) is organized under different packages

C. Create the Main Method and Print to Screen

- i. Double click your new class in the project window and it will be opened in the code editor on right panel
- ii. Enter the main method code and print out “Hello World!” to the console
- iii. Press **Ctrl-S** to save your code, and press **F6** to compile and run your project
- iv. Look at your project folder again, see where the bytecode file (*.class) is generated

D. Import Class from JDK and Get Input

- i. Enter the following line to your main method:
String value = JOptionPane.showInputDialog("Enter something:");
- ii. You will be warned that the JOptionPane class is not known. Press **Ctrl-Shift-I** to automatically import the required class. See what happen to your source.
- iii. Run your program again, and you will be prompted to enter a value.
- iv. What you typed in the popup dialogue will be stored in the string variable *value*.
- v. Print out the *value* to verify the result.

E. Two Useful Hotkeys

- i. When your code is not properly formatted or indented, try to press **Alt-Shift-F (Auto Format)**
- ii. When you forget the spelling of certain class/method/variable name, try to press **Ctrl-Space (Code Complete)**

F. Formatted Printing and Data Type Conversion

- i. Rewrite your main method as follow:

```
public static void main(String[] args) {  
    String value = JOptionPane.showInputDialog("Enter something:");  
    double halfValue = Double.parseDouble(value) / 2;  
    System.out.printf("Halve the value: %.2f\n", halfValue);  
}
```

- ii. Run the program and enter a number for testing
- iii. What is the use of command: `Double.parseDouble(...)`? (check API)
- iv. What is the use of `System.out.printf(...)`? (check API)
- v. What is the naming convention used by variable and method name?
- vi. What is the naming convention used by class name?

G. Set Breakpoint and Run Debugger

- i. In your code editor panel, you can set a breakpoint by clicking the line number located in its left hand side

```
14 public class NewClass {  
15     public static void main(String[] args) {  
16         String value = JOptionPane.showInputDialog("Enter something:");  
17         double halfValue = Double.parseDouble(value) / 2;  
18         System.out.printf("Halve the value: %.2f\n", halfValue);  
19     }  
20 }  
21
```

- ii. Now you can run your program in debug mode by pressing **Ctrl-F5**
- iii. After entering the value, your program will be suspended in the breakpoint

```
14 public class NewClass {  
15     public static void main(String[] args) {  
16         String value = JOptionPane.showInputDialog("Enter something:");  
17         double halfValue = Double.parseDouble(value) / 2;  
18         System.out.printf("Halve the value: %.2f\n", halfValue);  
19     }  
20 }
```

- iv. In the lower part of Netbeans, you will see the opened "Variables" window, in which you can inspect the variables in your code



- v. Press **F8** to step to the next line and inspect the halved value

```

14 public class NewClass {
15     public static void main(String[] args) {
16         String value = JOptionPane.showInputDialog("Enter something:");
17         double halfValue = Double.parseDouble(value) / 2;
18         System.out.printf("Halve the value: %.2f\n", halfValue);
19     }
20 }

```

- vi. You can press **Shift-F5** to finish debugging mode.

- Write a simple for-loop to print letters from A to Z.
- Given an array of integers, we say that a triple is a value appearing 3 times in a row in the array. Write a method *hasTriples()* that returns true if the array contains any triples. The method header is as follows:

```
boolean hasTriples(int[] nums)
```

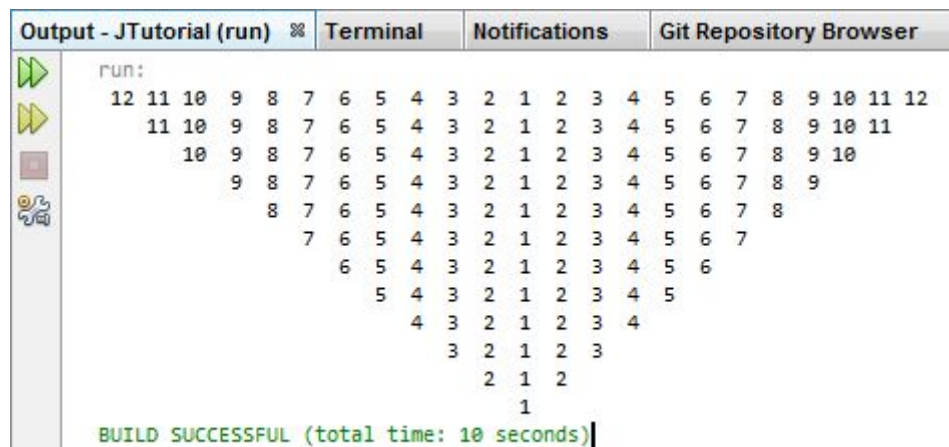
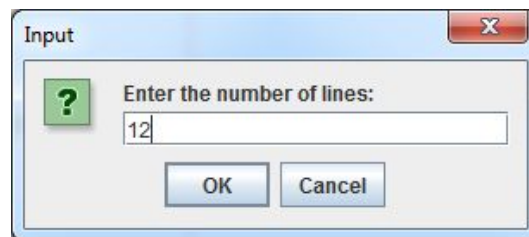
Ex.

{1, 1, 2, 2, 1} → false

{1, 1, 2, 2, 2, 1} → true

{1, 1, 1, 2, 2, 2, 1} → true

- [Checkpoint]** Write a class *PrintPyramid* that prompts the user to enter an integer ranging from 1 to 15 and displays a digit-pyramid. For example, if the input integer is 12, the output is shown below.



- END -