

CO225 – Software Construction -2017

Lab 0 – Introduction to NetBeans IDE

1. Command line Utilities

You can use the JDK command line utilities to compile and run Java programs. The JDK command line utilities consist of a set of separate programs, such as compiler and interpreter.

Open a text editor and type following program and save it as HelloWorld.java



```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        // Prints "Hello, World" to the terminal window.  
        System.out.println("Hello, World");  
    }  
  
}
```

Open a terminal and go to the location where you save HelloWorld.java. Use following command to compile the program.

```
$ javac HelloWorld.java
```

It will create HelloWorld. class file. Use following command to run the program

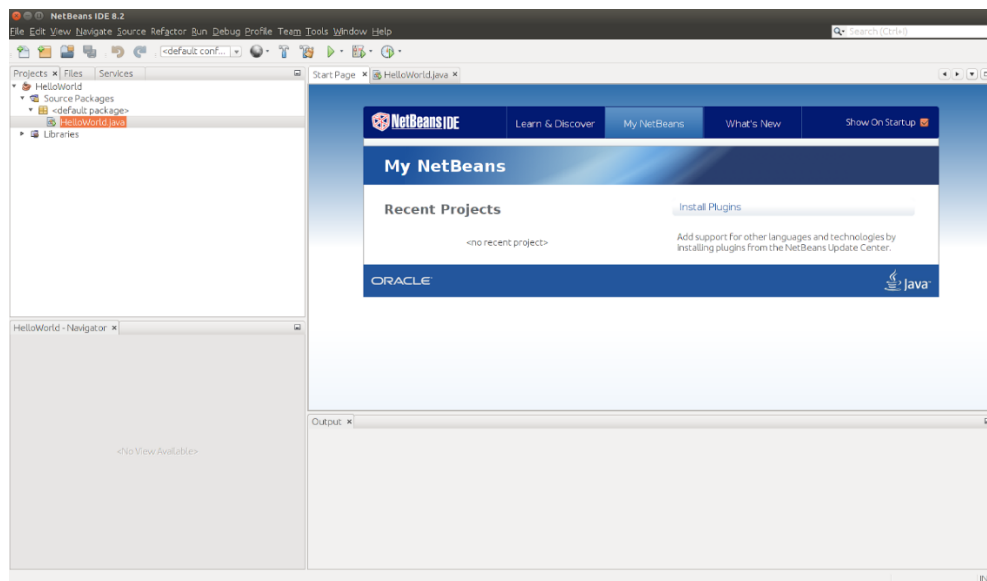
```
$ java HelloWorld
```

2. Why IDEs?

Integrated development environments (IDEs) are for rapid development of Java (or any other language) programs. Editing, compiling, building, debugging, online help, version controlling and many other things integrate in one graphical user interface. Using these tools effectively will greatly increase your programming productivity.

3. Getting Started with NetBeans

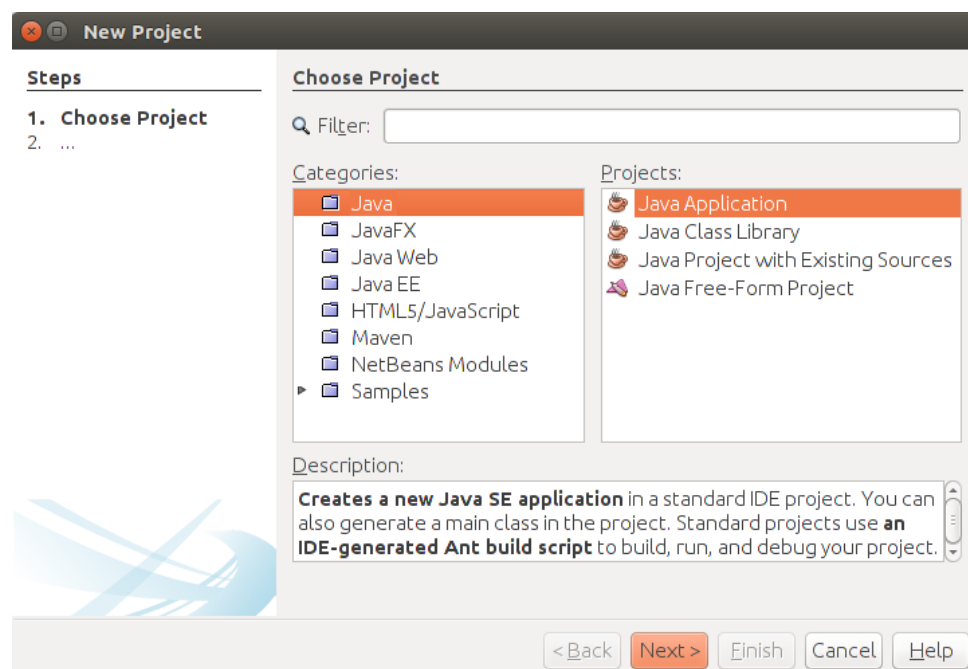
Open the NetBeans IDE. The NetBeans main window contains menus, toolbars and several panes.



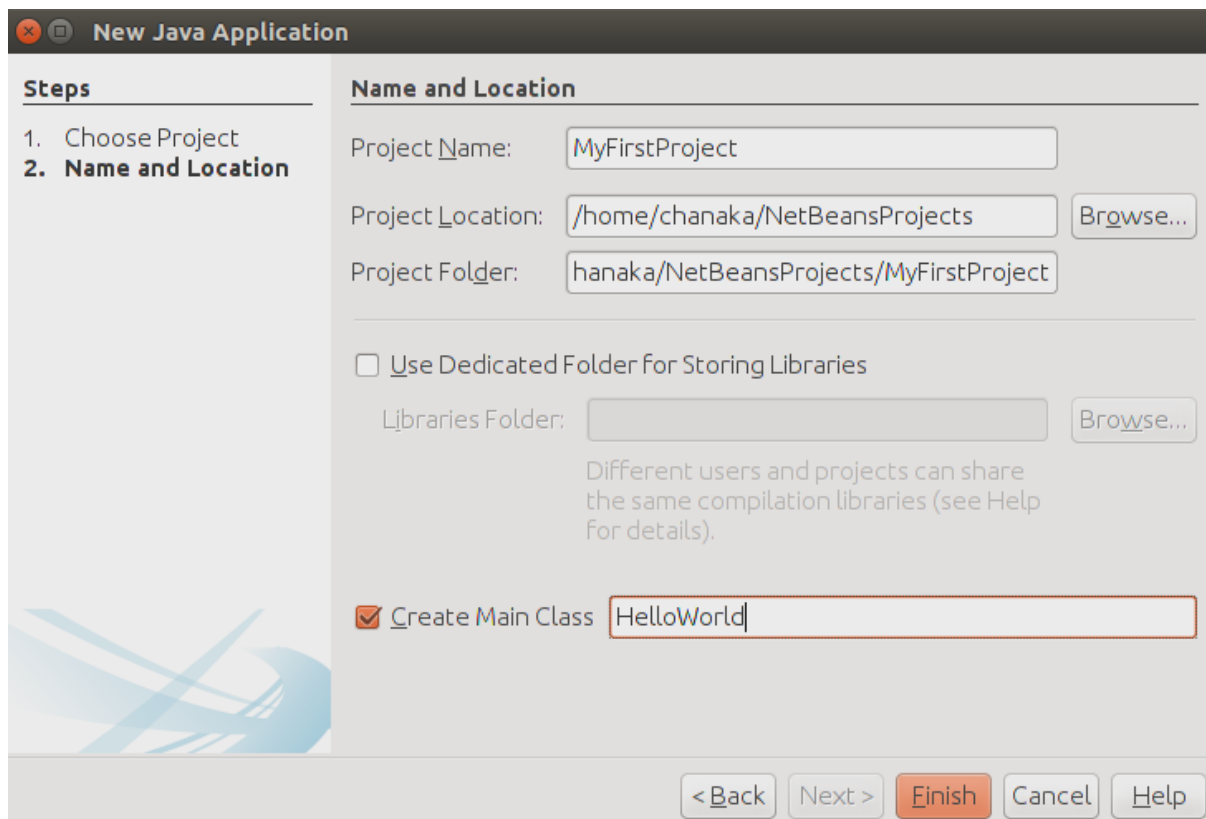
4. Creating a Project

A project contains information about programs and their dependent files, and it also stores and maintains the properties of the IDE settings. To create and run a program, you have to first create a project.

Choose File -> New Project to display the New Project dialog box



Select Java in the Categories section and Java Application in the Projects section and click Next to display the New Java Application dialog box.



Type “MyFirstProject” in the Project Name field and provide an appropriate location to save your project. You can create classes after a project is created. Optionally, you may also create the first class of the project when creating a new project. To do so, check the “Create Main Class” box and type the class name as “HelloWorld”. Click Finish to create the project.

Modify the code in HelloWorld class as shown in the figure below.

```
public class HelloWorld {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        System.out.println("Hello World !");  
    }  
}
```

TIP:

You can show line numbers in the Source Editor by choosing View -> Show Line Numbers from the main menu.

5. Creating a Class

You can create multiple classes in a project. Right-click on the top node of your project in the project pane to display a context menu. Choose “New Java Class” to display the New Java Class dialog.

New Java Class

Steps

1. Choose File Type
2. Name and Location

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

⚠ Warning: It is highly recommended that you do not place Java classes in the default package.

< Back Next > **Finish** Cancel Help

Type “Welcome” in the Class Name field and select the “Source Packages” in the Location field. Leave the Package field blank. This will create a class in the default package. (Note that there is a warning and it is not recommended to use the default package, but it is fine for now. You will later learn about packages and to use classes in a non-default packages.)

Modify the code in Welcome class as shown in the figure below

```
public class Welcome {  
    public static void main(String[] args) {  
        // Display message "Welcome to C0225"  
        System.out.println("Welcome to C0225 !");  
    }  
}
```

6. Compiling a Class

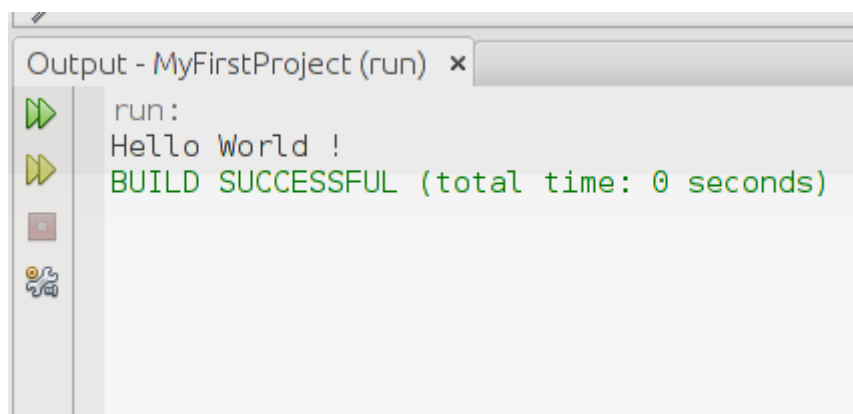
To compile Welcome.java, right-click Welcome.java to display a context menu and choose Compile File, or simply press F9 when the Welcome.java is active in the Editor pane. The compilation status is displayed in the Output pane. If there are no syntax errors, the compiler generates a file named Welcome.class, which is stored in <your project location>\build\classes. Look at this location & see what files are there.

The screenshot shows the 'Output - MyFirstProject (compile-single)' window. It contains the following text: 'ant -f /home/chanaka/NetBeansProjects/MyFirstProject -Djavac.includes=Welcome.jav', 'init:', 'deps-jar:', 'Created dir: /home/chanaka/NetBeansProjects/MyFirstProject/build', 'Updating property file: /home/chanaka/NetBeansProjects/MyFirstProject/build/built', 'Created dir: /home/chanaka/NetBeansProjects/MyFirstProject/build/classes', 'Created dir: /home/chanaka/NetBeansProjects/MyFirstProject/build/empty', 'Created dir: /home/chanaka/NetBeansProjects/MyFirstProject/build/generated-source', 'Compiling 1 source file to /home/chanaka/NetBeansProjects/MyFirstProject/build/cl', 'compile-single:', and 'BUILD SUCCESSFUL (total time: 1 second)'. On the left side of the window, there are four icons: a green play button, a yellow play button, a red stop button, and a magnifying glass icon.

```
ant -f /home/chanaka/NetBeansProjects/MyFirstProject -Djavac.includes=Welcome.jav
init:
deps-jar:
Created dir: /home/chanaka/NetBeansProjects/MyFirstProject/build
Updating property file: /home/chanaka/NetBeansProjects/MyFirstProject/build/built
Created dir: /home/chanaka/NetBeansProjects/MyFirstProject/build/classes
Created dir: /home/chanaka/NetBeansProjects/MyFirstProject/build/empty
Created dir: /home/chanaka/NetBeansProjects/MyFirstProject/build/generated-source
Compiling 1 source file to /home/chanaka/NetBeansProjects/MyFirstProject/build/cl
compile-single:
BUILD SUCCESSFUL (total time: 1 second)
```

7. Running a Java Application

To run HelloWorld.java, right-click on HelloWorld.java to display a context menu and choose Run File, or simply press Shift + F6. The output is displayed in the Output pane.

The screenshot shows the 'Output - MyFirstProject (run)' window. It contains the following text: 'run:', 'Hello World !', and 'BUILD SUCCESSFUL (total time: 0 seconds)'. On the left side of the window, there are four icons: a green play button, a yellow play button, a red stop button, and a magnifying glass icon.

```
run:
Hello World !
BUILD SUCCESSFUL (total time: 0 seconds)
```

Note: If you run a file without compiling it, the compilation step will be done in the background. Check <your project location>\build\classes after running the HelloWorld.java

8. Adding external source codes to a project

Download InfiniteLoop.java file and copy it in to <your project location>/src directory. The class will be appearing in your project under default package.

9. Forcing a program to terminate

If a program does not terminate due to a logic error, you can force it to terminate by clicking the Stop icon in the output pane.

Run InfiniteLoop.java and force to terminate it.

TIP: You can see labels known as ToolTip for a button by pointing the mouse to the button without clicking it.

10. Debugging in NetBeans

The debugger utility is integrated with NetBeans. The NetBeans debugger enables you to set breakpoints and execute programs line by line. As your program executes, you can watch the values stored in variables, observe which methods are being called, and know what events have occurred in the program. Apart from error detection it helps you understand how a program is executed.

10.1 Setting Breakpoints

You can execute a program line by line to trace it. But this is time-consuming if you are debugging a large program. Often, you know that some parts of the program work fine. It makes no sense to trace these parts when you only need to trace the lines of code that are likely to have bugs. In cases of this kind, you can use breakpoints.

A breakpoint is a stop sign placed on a line of source code that tells the debugger to pause when this line is encountered. The debugger executes every line until it encounters a breakpoint. You can then trace the part of the program at the breakpoint, quickly moving over the sections that work correctly and concentrating on those causing problems.

To set a breakpoint, click on the left side margin of the line (where line numbers are displayed) on which you want to put a breakpoint. You will see the line highlighted. You also can set breakpoints by choosing Debug-> New Breakpoint. To remove a breakpoint, simply click again on the line.

Download DebugMe.java file and copy it into <your project location>/src directory. Run it to see how it is working.

Then set two break points on line number 16 and 18.

```
11 public class DebugMe {
12     public static void main (String[] args)
13     {
14         for ( int i = 1 ; i< 20 ; i++)
15         {
16             System.out.println("count =" +i);
17             System.out.println("-----");
18             ShowCurrentTime();
19         }
20     }
21
22     public static void ShowCurrentTime()
23     {
24         // Obtain the total milliseconds since the midnight, Jan
25         long totalMilliseconds = System.currentTimeMillis();
26
27         // Obtain the total seconds since the midnight, Jan 1, 19
28         long totalSeconds = totalMilliseconds / 1000;
29
30         // Compute the current second in the minute in the hour
31         int currentSecond = (int)(totalSeconds % 60);
32     }
33 }
```

10.2 Starting the debugger

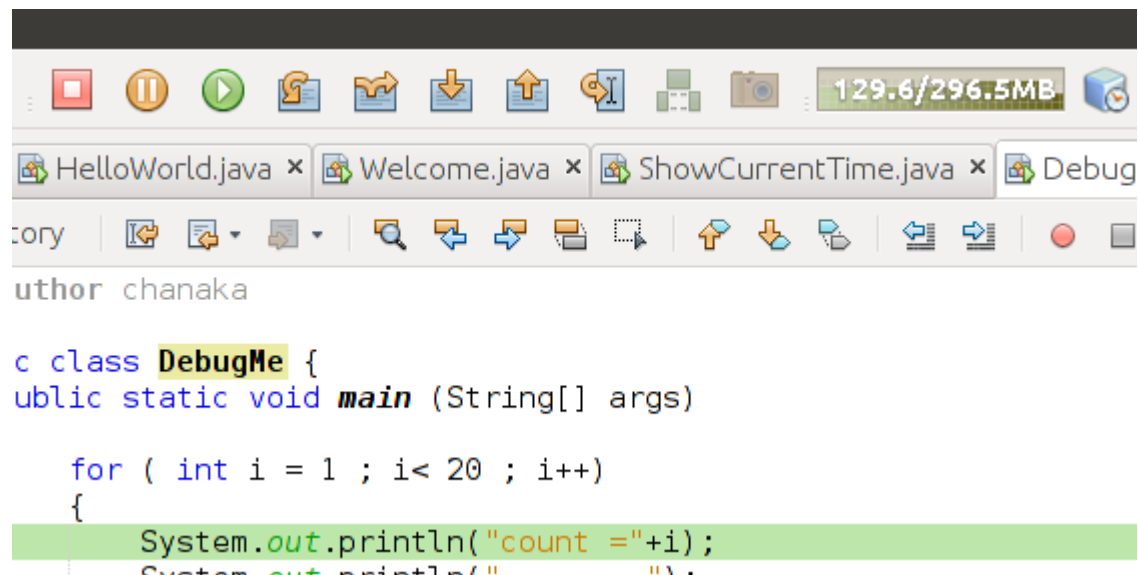
Right click on the DebugMe.java on project pane and in the context menu, choose “Debug File”. If the program compiles properly, an output pane and the debug pane will be displayed, as shown below. If the debug pane is not shown, choose Window -> Debugging-> Variables to display it. To display the call stack & watches tabs choose Window -> Debugging-> Call Stack / Watches.



10.3 Controlling program execution

The program pauses at the first breakpoint. The paused line is highlighted in green. This line is the next statement to be executed by the debugger. When the program pauses, you can issue debugging commands to control the execution of the program. You also can inspect or modify the values of variables in the program.

When NetBeans is in the debugging mode, the toolbar buttons for debugging are displayed. The toolbar button commands also appear in the Debug menu.



Here are some commands for controlling program execution.

Finish - ends the current debugging session.

Pause - temporarily stops execution of a program.

Step Over - executes a single statement. If the statement contains a call to a method, the entire method is executed without stepping through it.

Step Into - executes a single statement or steps into a method.

Step Out - executes all the statements in the current method and returns to its caller.

Run to Cursor - runs the program, starting from the current execution point, and pauses and places the execution point either in the line of code containing the cursor or at a breakpoint.

i. Place your cursor at line 17 and check how Run to Cursor works.

ii. Check how Step Over works.

iii. Step into ShowCurrentTime() method and see the call stack.

iv. Step out from ShowCurrentTime() method and see the call stack.

While doing above tasks, observe the variables and output tabs in the debug pane. In the output tab, there are two tabs for console output, and for debugger console. You may finish the current debugging session and start a debugging session once again if required.

10.4 Examining and Modifying Data Values

You can examine the values of variables, array items, and objects, or the values of the parameters passed in a method call. You also can modify a variable value if you want to try a new value to continue debugging without restarting the program.

Start a new debugging session and step over several times (till $i=3$). Now on the variables tab, click on the value of “i” and set it to 0. Step over once again and see the output.

10.5 Adding watches

Using watches you can observe how the value of variables and expressions are changing.

Start a new debugging session. Highlight the variable *i* and right click. In the context menu, select “New Watch”. Keep watch expression as ‘i’ and click OK.

You can also add expressions to watches. Add another three watches for expression ‘ $i < 3$ ’, ‘ $i < 20$ ’ and ‘ $i+10$ ’

Step over through the program and on the watches tab observe how the values of the above expressions changing

11. Netbeans Help

NetBeans provides a large number of help documents, giving you a great deal of information on a variety of topics. Choose Help -> Help Contents to display NetBeans Help.

