



جامعة الأمير محمد بن فهد الأهلية  
PRINCE MOHAMMAD BIN FAHD UNIVERSITY

**COMM 2312 Technical and Professional Communication**

**Dr. Jörg Muth**

**Fall 2019**

**Class section 107**

**Assignment 3: Creation of a Technical Manual**

<b>Abdulrahman E. S Aleid</b>	<b>201800290</b>
<b>Waleed R. Alhindi</b>	<b>201800509</b>
<b>Ali Albrikan</b>	<b>201601788</b>



# NetBeans

## Coding Java with Netbeans

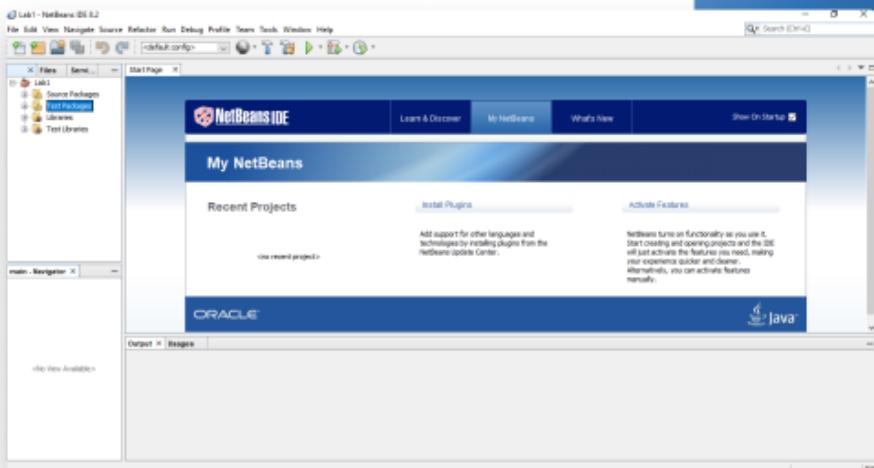
### Starter's manual



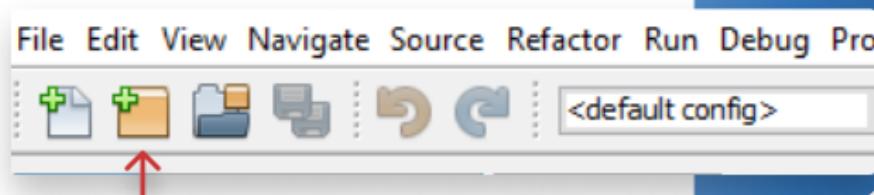
# How to create a project

## Step 1:

The first step to coding Java in Netbeans is starting a project. Open Netbeans and you should see the following page:



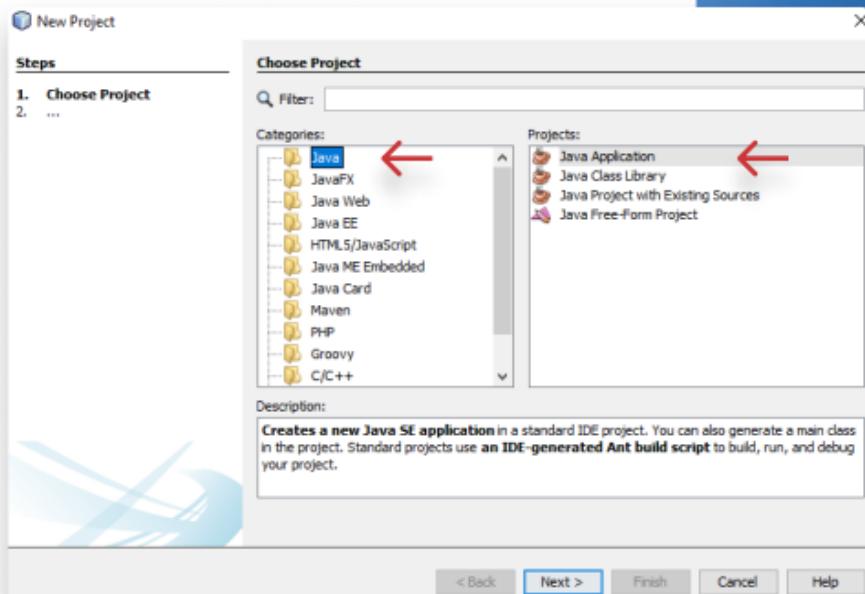
Then, press on the “New Project” icon . It should appear as the second left-most icon on the toolbar icons located at the top as shown below:



# 2

## Step 2:

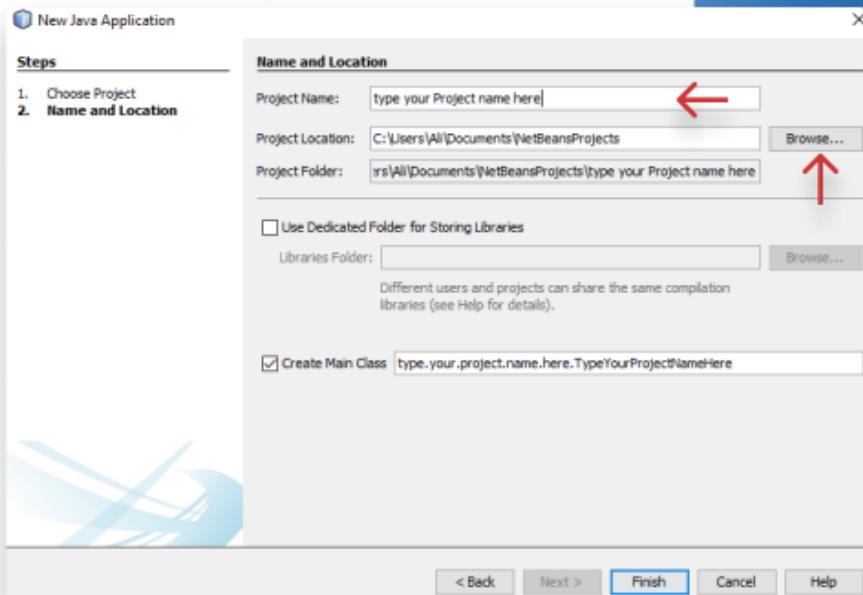
When you press the icon, this window will pop up. Select “Java” from the “Categories” list on the left. A new list will open on the right called “Projects”. Select “Java Applications” from the list of “Projects” that appears on the right as shown below:



# 3

## Step 3:

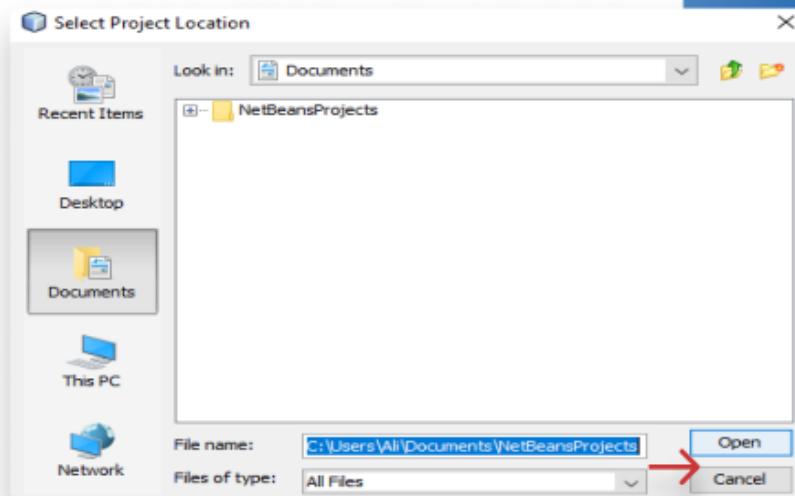
Now press the “Next” button on the bottom of the window. A new window will now appear that asks you to name your project. Enter your desired project name in the first text-box labeled “Project name” as shown below:



# 4

## Project memory location:

You can choose where to save the project in your computer by clicking the “Browse” button next to the “Project Location” text-box. This will produce a new pop-up window that will ask you where you want to store the project on your computer’s memory.



Choose a location then press “Open” on the bottom right. This brings you back to the previous window.

# 5

Finally, press “Finish” at the bottom of the window.



< Back

Next >

Finish

Cancel

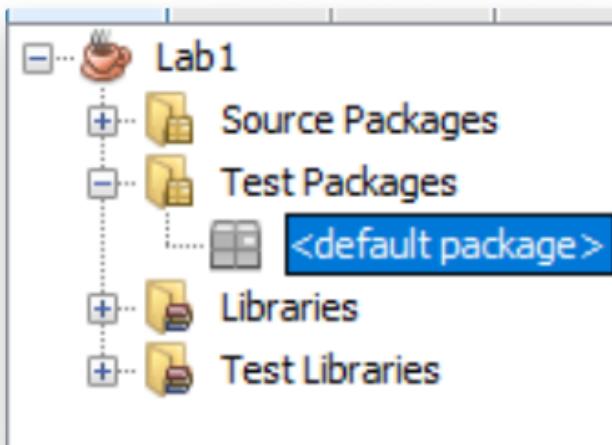
Help

Now, you have created your own project. The window shown below is where you will be developing Java code:

```
1 // To change this license header, choose License Headers in Project Properties.
2 // To change this template file, choose Tools | Templates
3 // and open the template in the editor.
4 //
5 package type.your.project.name.here;
6
7 /**
8 * Author All
9 */
10 public class TypeYourProjectNameHere {
11
12     /**
13      * Spares args the command line arguments
14      */
15     public static void main(String[] args) {
16         // TODO code application logic here
17     }
18 }
```

# Navigating Netbeans

## Source code\* editing window:



The window above contains:

- **The project's package:**  
a collection of the classes contained in your project.
- **The public class:**  
contains main and non-main methods and classes.
- **The main method:**  
this is where you will be writing your Java application code. This is demonstrated by the basic print method inside the main method's section.

## Output (console) window:

The console is where your Java code from the source code window is compiled and displayed.

A screenshot of a Java IDE's source code editor. The code is as follows:

```
9  *
10 * Author Ali
11 */
12 public class Exampleprogram {
13
14     /**
15      * @param args the command line arguments
16     */
17    public static void main(String[] args) {
18        System.out.println("Hello World");
19    }
20
21
22 }
```

A red curly brace on the right side of the code highlights the entire block of code, with the text "Source Code" written next to it.

The picture below demonstrates the above's source code's compiled output displayed onto the console window:

A screenshot of a Java IDE's output window titled "Output - exampleprogram [run]". It shows the following text:

```
INFO
Hello WorldBUILD SUCCESSFUL (total time: 0 seconds)
```

A red curly brace on the right side of the output text highlights the "Hello World" line, with the text "Output" written next to it.

## Toolbar:



-  New file: Creates a new file within the package of a project.
-  New project: Creates a new project (as shown in the first section of the manual).
-  Open project: Opens a Netbeans project that you already created.
-  Save all: Saves all projects you currently have open on Netbeans.
-  Undo: Reverts (undos) your last action on Netbeans.
-  Redo: Reverts your most recent deletion in the source code window.

<default config> Project configuration: allows you to customize the specifics of your project (i.e. Main class, arguments, runtime platform, etc).

 Build projects: runs all projects currently opened in, or currently referenced in, your Netbeans application.

 Clean and build projects: debugs and runs all projects currently opened in, or currently referenced in, your Netbeans application.

 Run project: runs the source code within the project you have open in the source code editing window.

 Debug project: debugs the source code within the project you have open in the source code editing window.

- Profile project: provides the properties of the elements within the project you have open in the source code editing window.

---

\*Source code: a high-level written language, such as Java, that tells Netbeans what to execute when a program (project) is run.

---

# Common Errors

## Types of errors:

- Bugs: A mistake in the program.  
(Note: getting rid of errors is called “debugging”).
- Syntax error: A grammatical mistake in the formation of the code. Netbeans will automatically detect syntax errors as you write code and will often suggest fixes.
- Runtime error: An error that will only be detected as the program is run. The error message is displayed in the output window after the program is executed.

-Logic errors: A mistake in the logic behind a program like creating a loop without an endpoint. Logic errors are not detected while coding nor while running the program. So, no error message will arise because Netbeans cannot assess the underlying logic of a program.

### **Identifier errors:**

Identifiers: the name a user gives to variables, methods, classes and objects. The user defines these identifiers within the source code of a program (project).

## Restrictions when naming identifiers:

- An identifier cannot start with a digit.
- An identifier cannot contain any characters other than letters, digits, and the underscore symbol.
- Identifiers can be of any length (no limit on the amount of characters of an identifier).
- Identifiers are case-sensitive. So, for example, the identifiers "example", "Example" and "EXAMPLE" will be considered as three separate identifiers.

-Identifiers cannot be any of Java's reserved words. Reserved words are outlined in the following section.

### Reserved words:

Reserved words are words that Java internally uses as identifiers. Thus, these words cannot be used when as any user-generated identifiers. The following are Java's reserved words:

abstract	default	if
assert	do	implements
boolean	double	import
break	else	instanceof
byte	extends	int
case	false	interface
catch	final	long
char	finally	native
class	float	new
const	for	null
continue	goto	

private	throw
protected	throws
public	transient
return	true
short	try
static	void
strictfp	volatile
super	while
switch	
synchronized	
this	

**Note:** be cautious of using the words in the table above as the name of your variables, methods and/or classes.