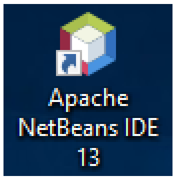
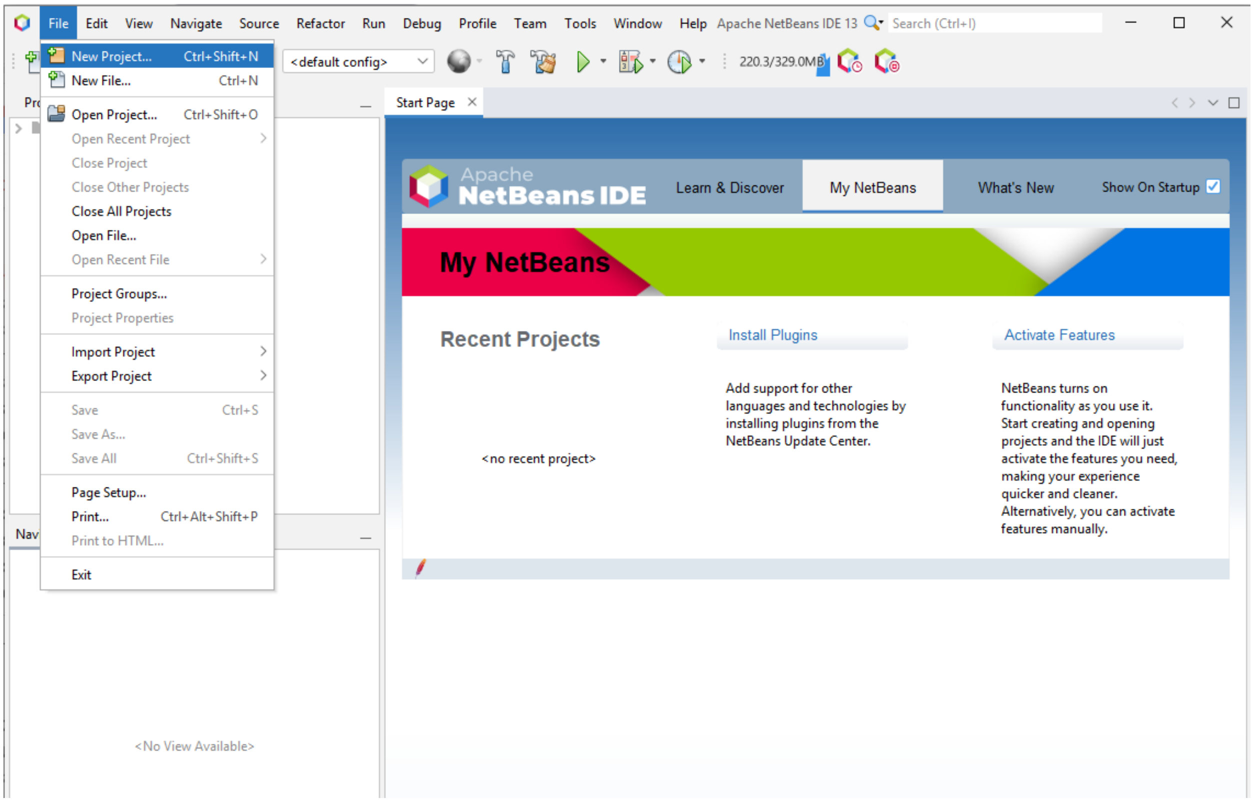


Click on the NetBeans IDE icon to begin.

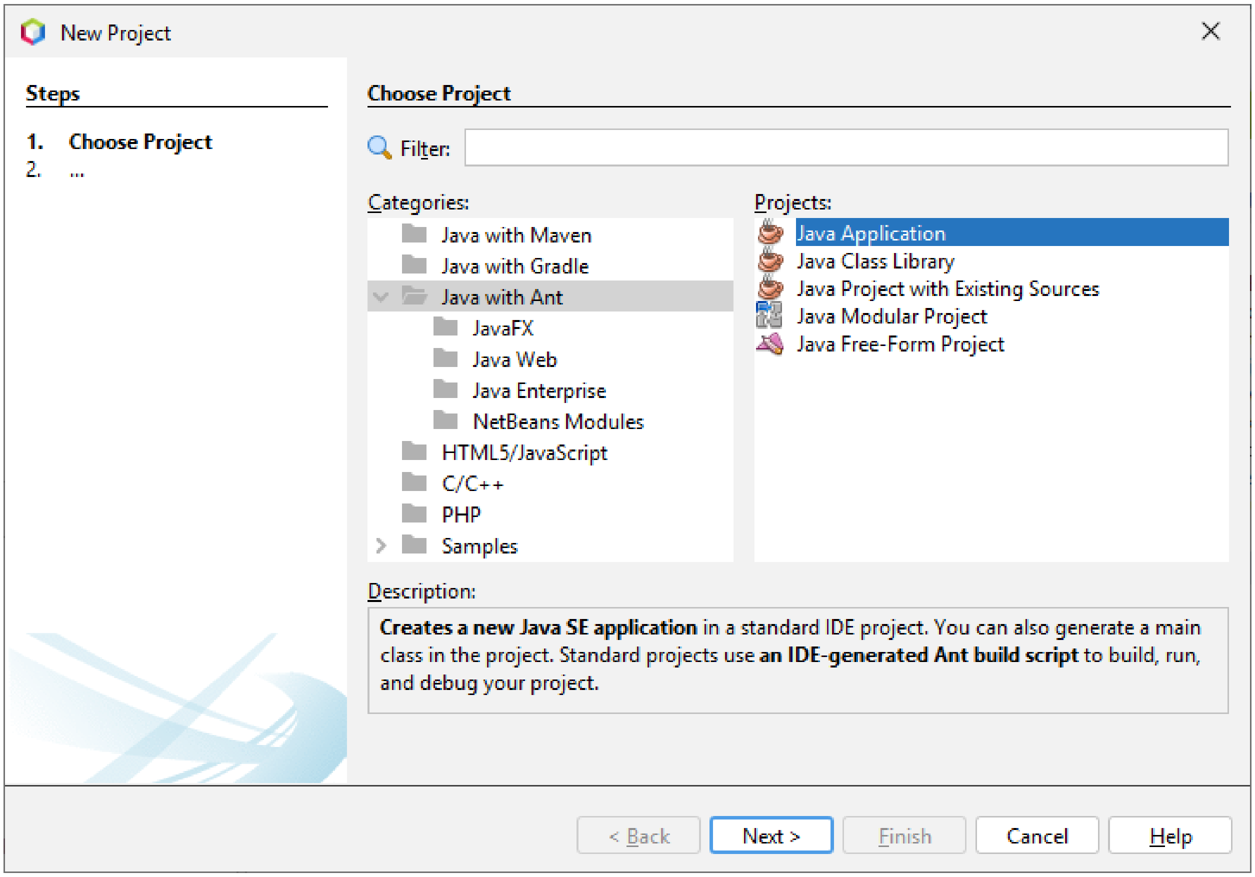
Making a New Project



Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.

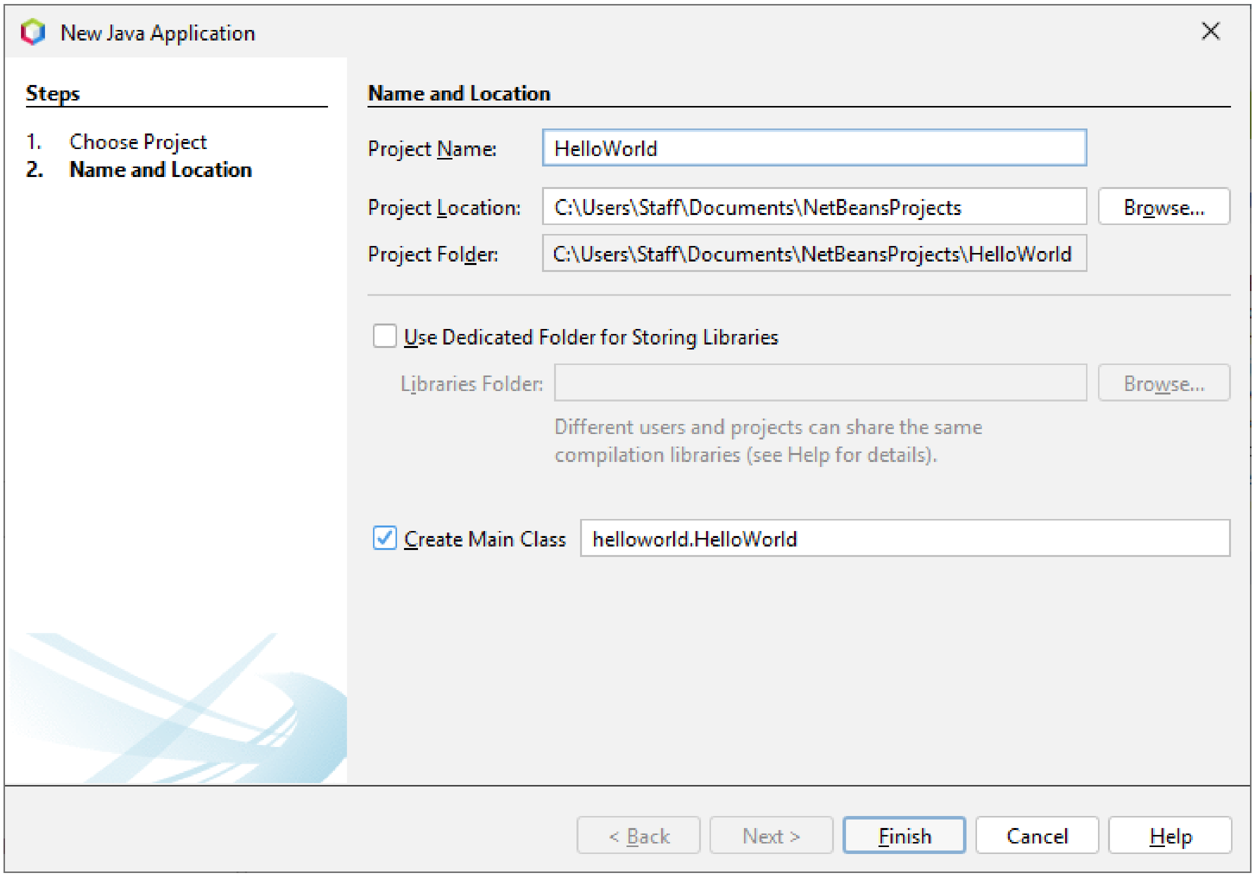
From the File menu, select **New Project**.



Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

From the Categories menu select **Java with Ant**. From the Projects menu select **Java Application**. Then press **Next**.

New Java Application



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: HelloWorld

Project Location: C:\Users\Staff\Documents\NetBeansProjects

Project Folder: C:\Users\Staff\Documents\NetBeansProjects\HelloWorld

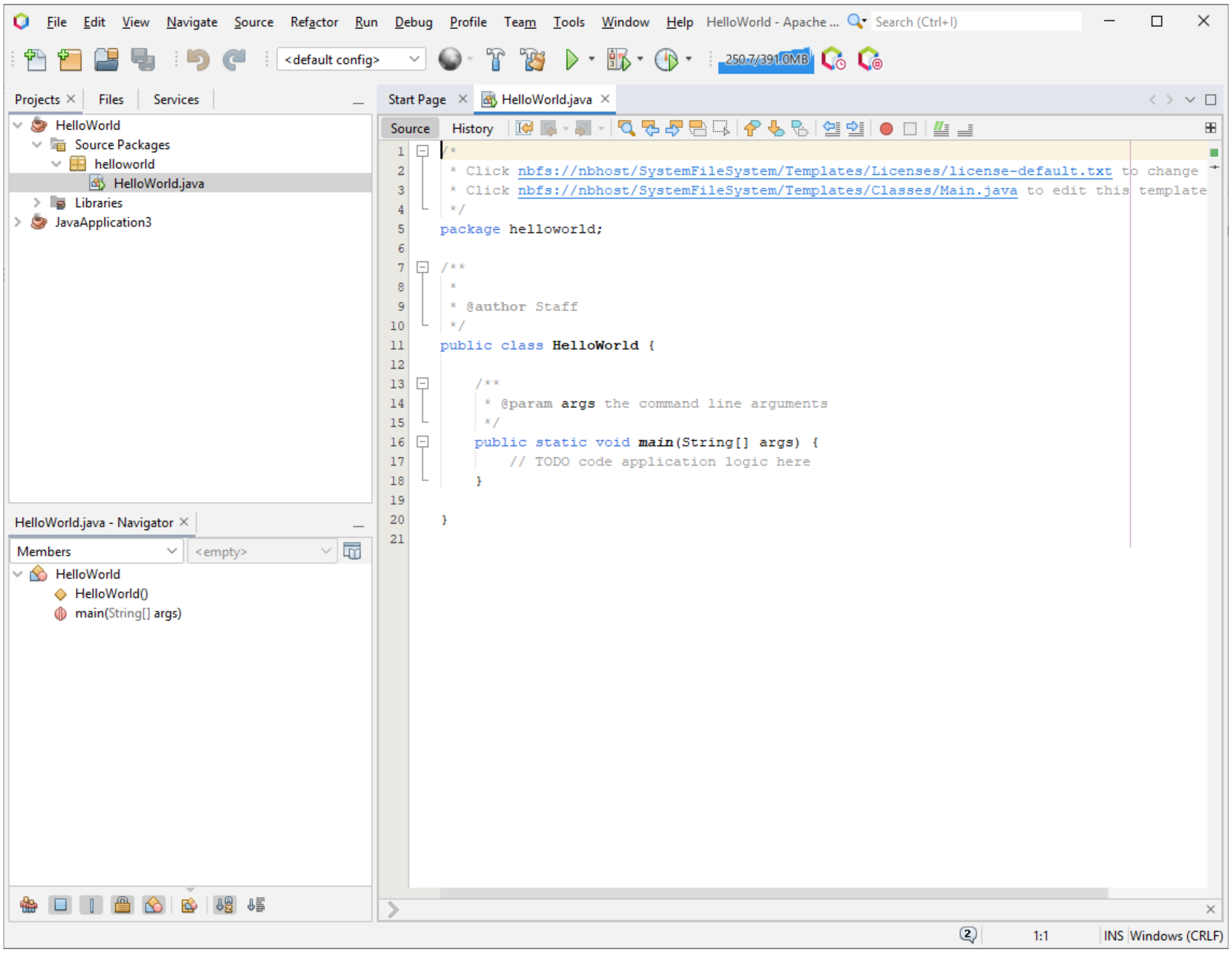
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class helloworld.HelloWorld

The Project Name typed was HelloWorld. The NetBeans generated the rest of the values (including the Create Main Class). Click on Finish when done. It will take a few seconds as the project is created.

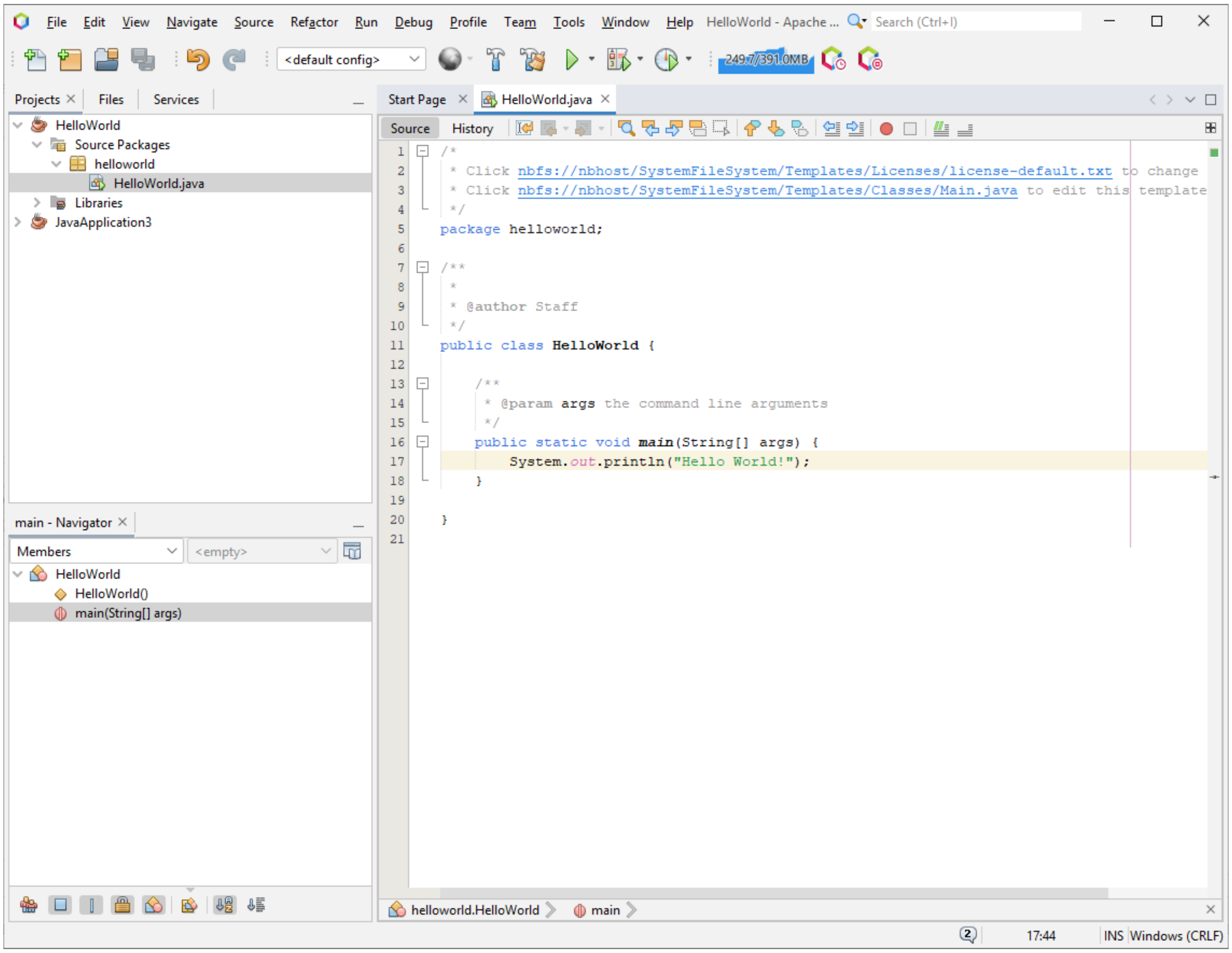


```
1   /*
2    * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
3    * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4    */
5   package helloworld;
6
7   /**
8    *
9    * @author Staff
10   */
11  public class HelloWorld {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18      }
19
20  }
21
```

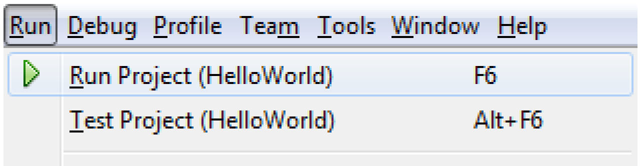
Here is a blank project. We will now fill in some code to make the program do something. Replace line 17 with the following:

System.out.println("Hello World!");

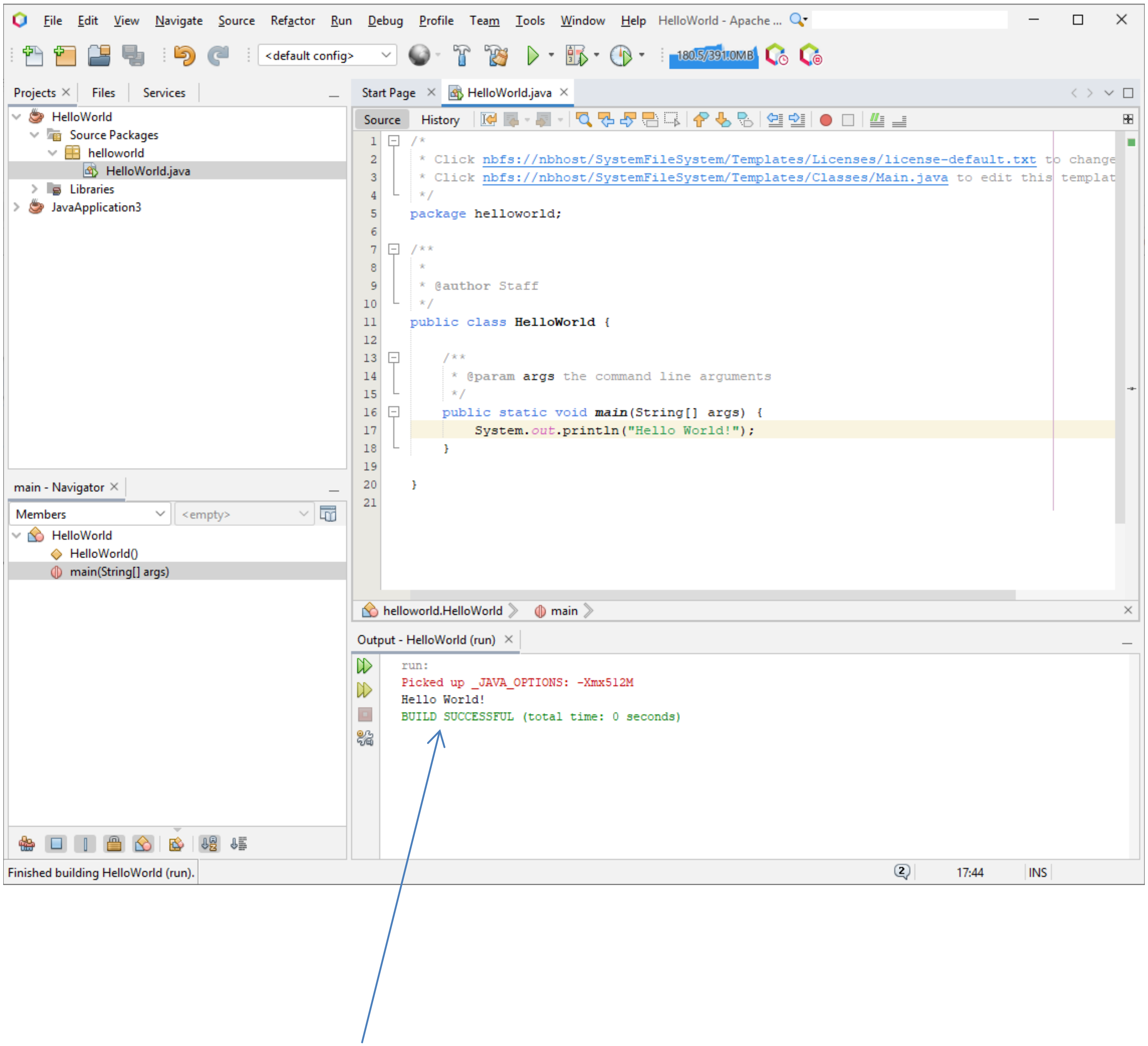
Now we have a new program:



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package helloworld;

/**
 *
 * @author Staff
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}
```


Run Debug Profile Team Tools Window Help
Run Project (HelloWorld) F6
Test Project (HelloWorld) Alt+F6

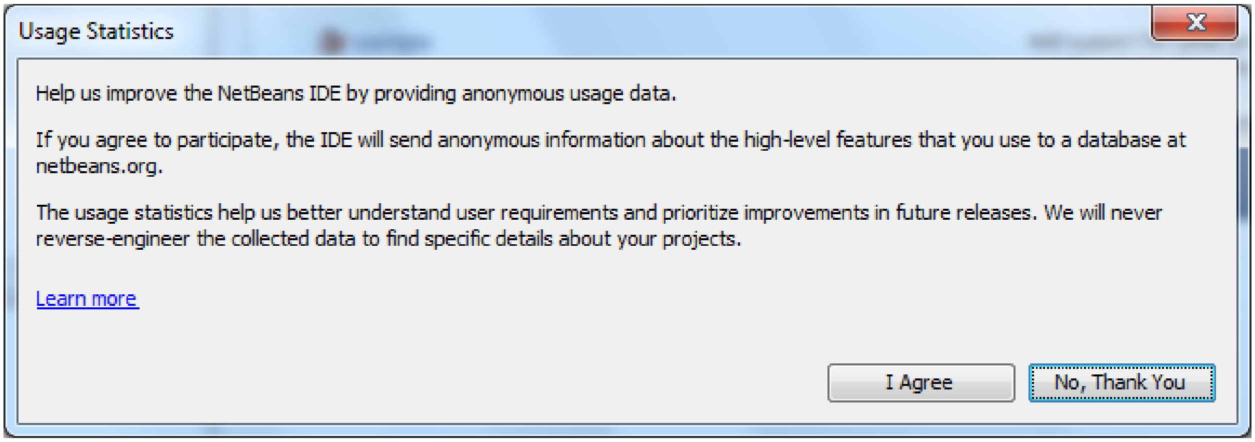
From the Run menu, select Run Project. You could press the keyboard shortcut F6, if you prefer.



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this templat
 */
package helloworld;

/**
 *
 * @author Staff
 */
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}
```

```
run:
Picked up _JAVA_OPTIONS: -Xmx512M
Hello World!
BUILD SUCCESSFUL (total time: 0 seconds)
```

As shown above, the output from the program is displayed in the Output window in NetBeans.



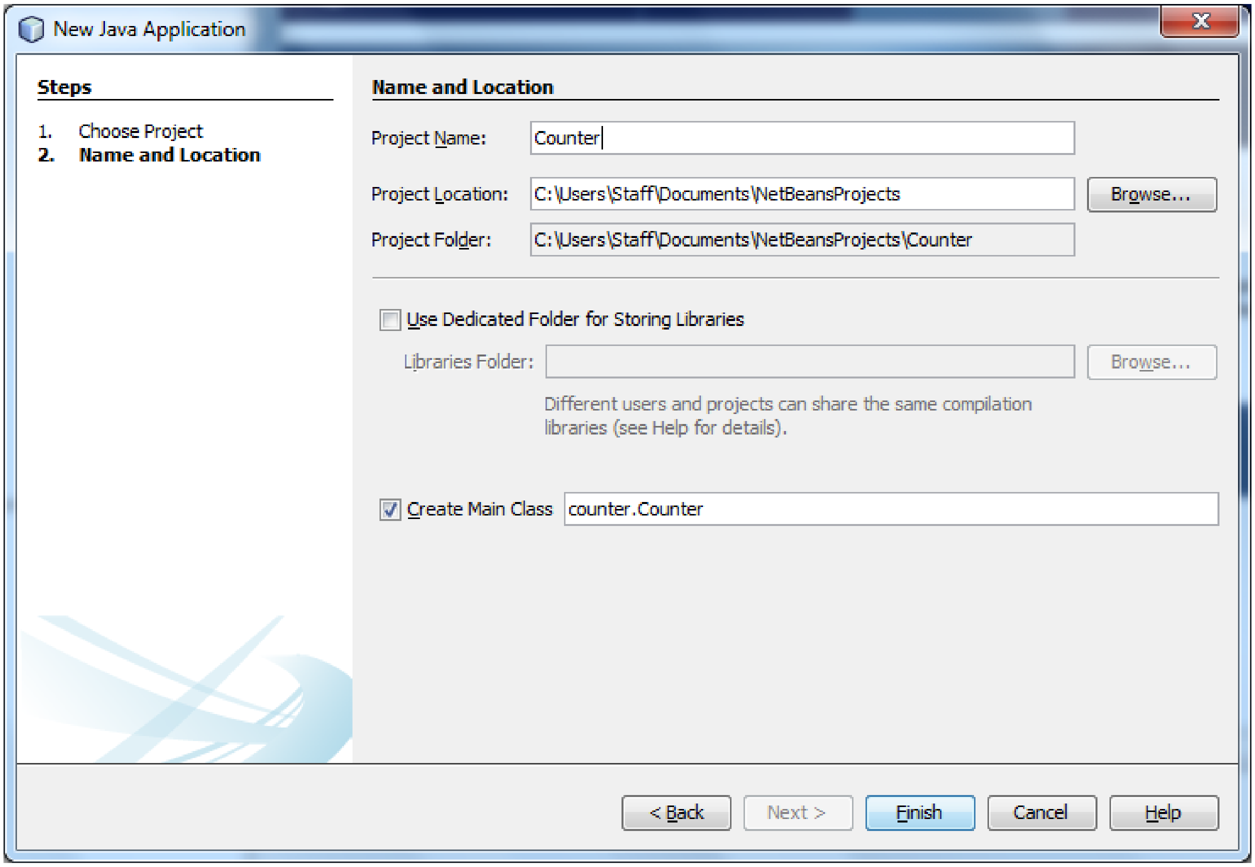
Help us improve the NetBeans IDE by providing anonymous usage data.

If you agree to participate, the IDE will send anonymous information about the high-level features that you use to a database at netbeans.org.

The usage statistics help us better understand user requirements and prioritize improvements in future releases. We will never reverse-engineer the collected data to find specific details about your projects.

A popup window about Usage Statistics may appear when you start NetBeans. If you would like to participate in anonymous information being sent to NetBeans to help them improve the program, click on I Agree. Otherwise, you may wish to say **No, Thank You**.

Let’s make a more complicated program. We will make a New Java Application. The Project Name will be called Counter. Check on Finish when done.



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: Counter

Project Location: C:\Users\Staff\Documents\NetBeansProjects

Project Folder: C:\Users\Staff\Documents\NetBeansProjects\Counter

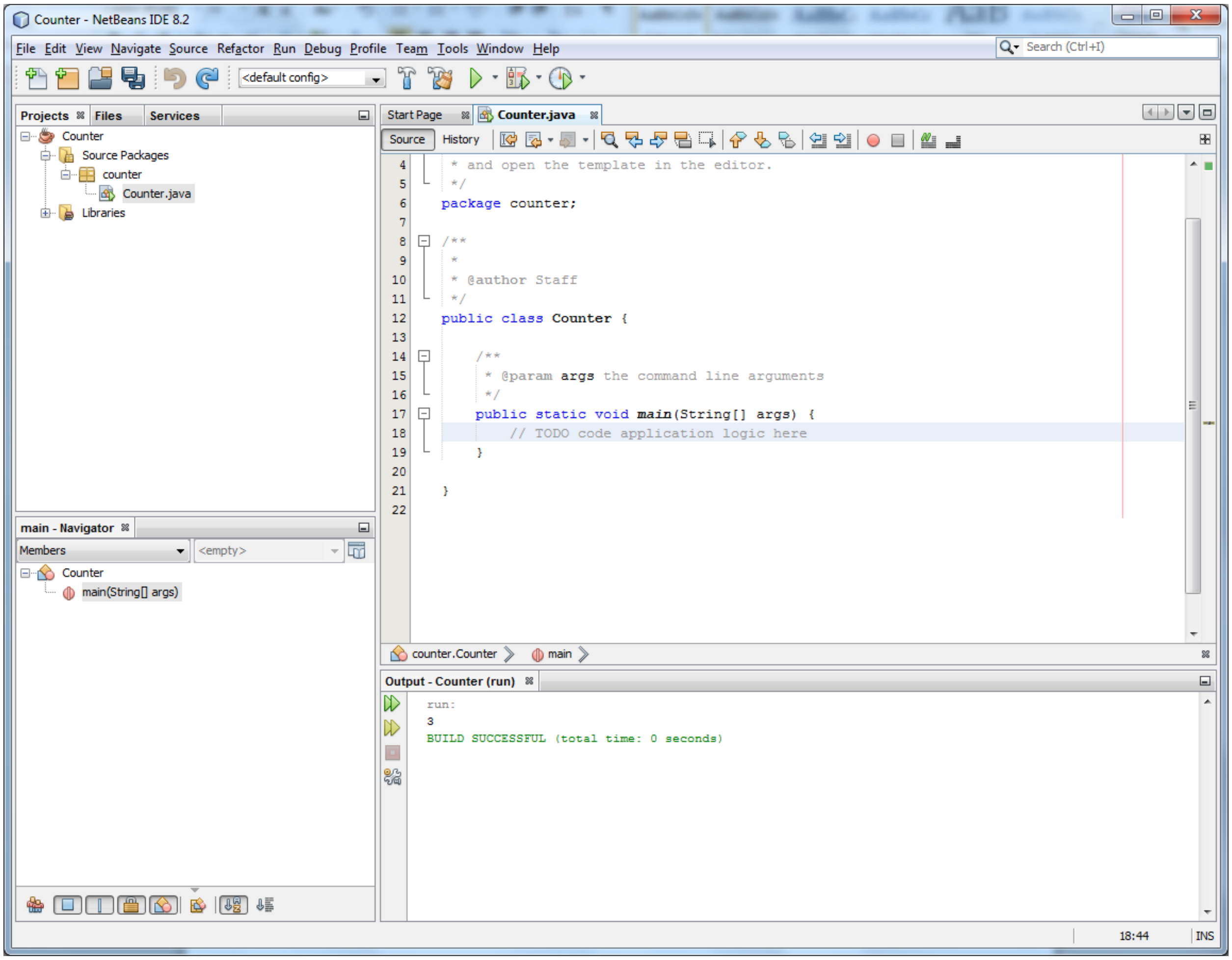
□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

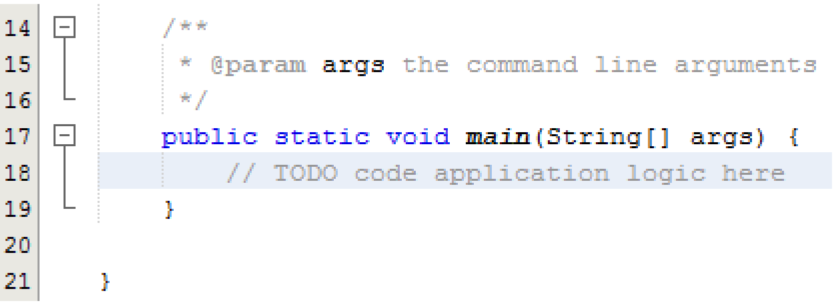
☑ Create Main Class counter.Counter

Now you have a Java Program created with a skeleton code for you to fill in:



```
 4   * and open the template in the editor.
 5   */
 6  package counter;
 7
 8  /**
 9   *
10   * @author Staff
11   */
12  public class Counter {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20
21  }
22
```

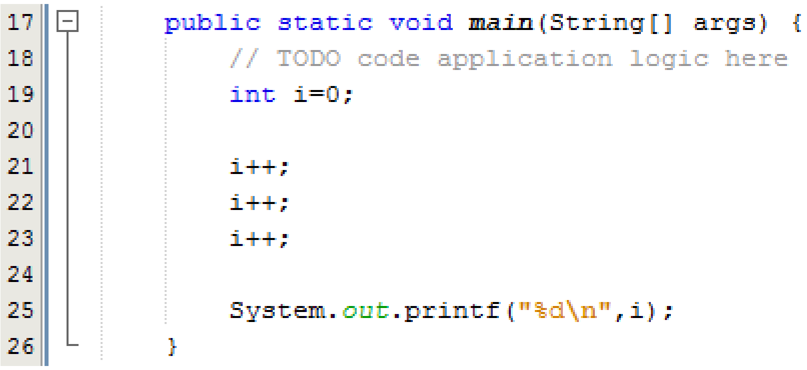
```
run:
3
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20
21  }
```

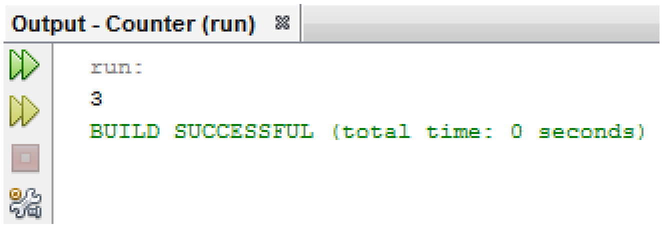
We will begin our program at line 19:

```
int i=0;

i++;
i++;
i++;

System.out.printf("%d\n",i);
```



```
17      public static void main(String[] args) {
18          // TODO code application logic here
19          int i=0;
20
21          i++;
22          i++;
23          i++;
24
25          System.out.printf("%d\n",i);
26      }
```

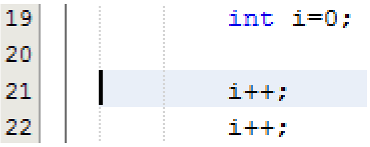
If we Run the program (Press F6), we see the following output:



```
run:
3
BUILD SUCCESSFUL (total time: 0 seconds)
```

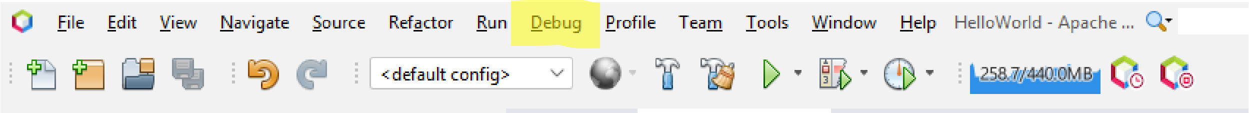
The value of i is displayed on the screen. But how did it become the value of 3. Let us debug the program to see what happens to the variables as the program runs.

Select the location the source code that you would like the program to stop at.

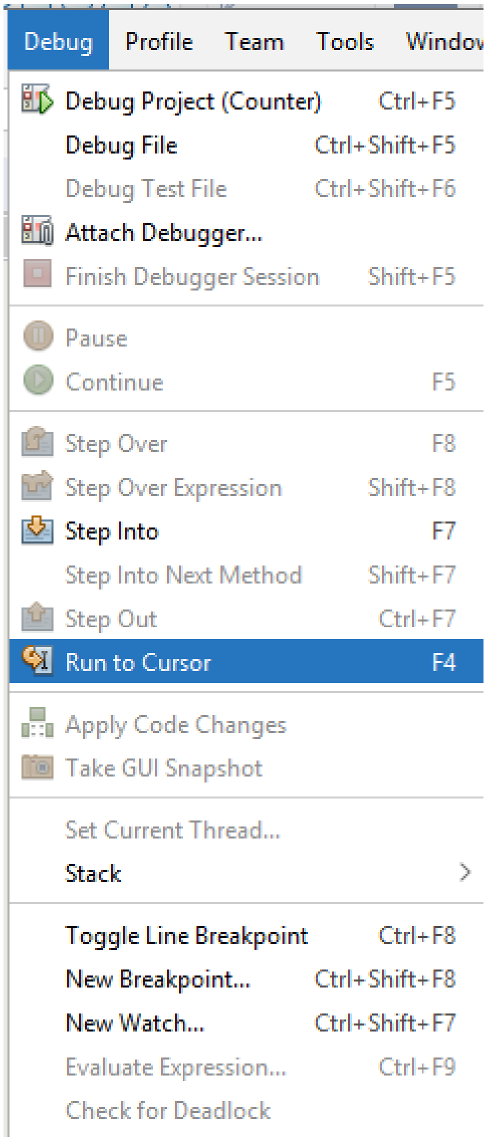


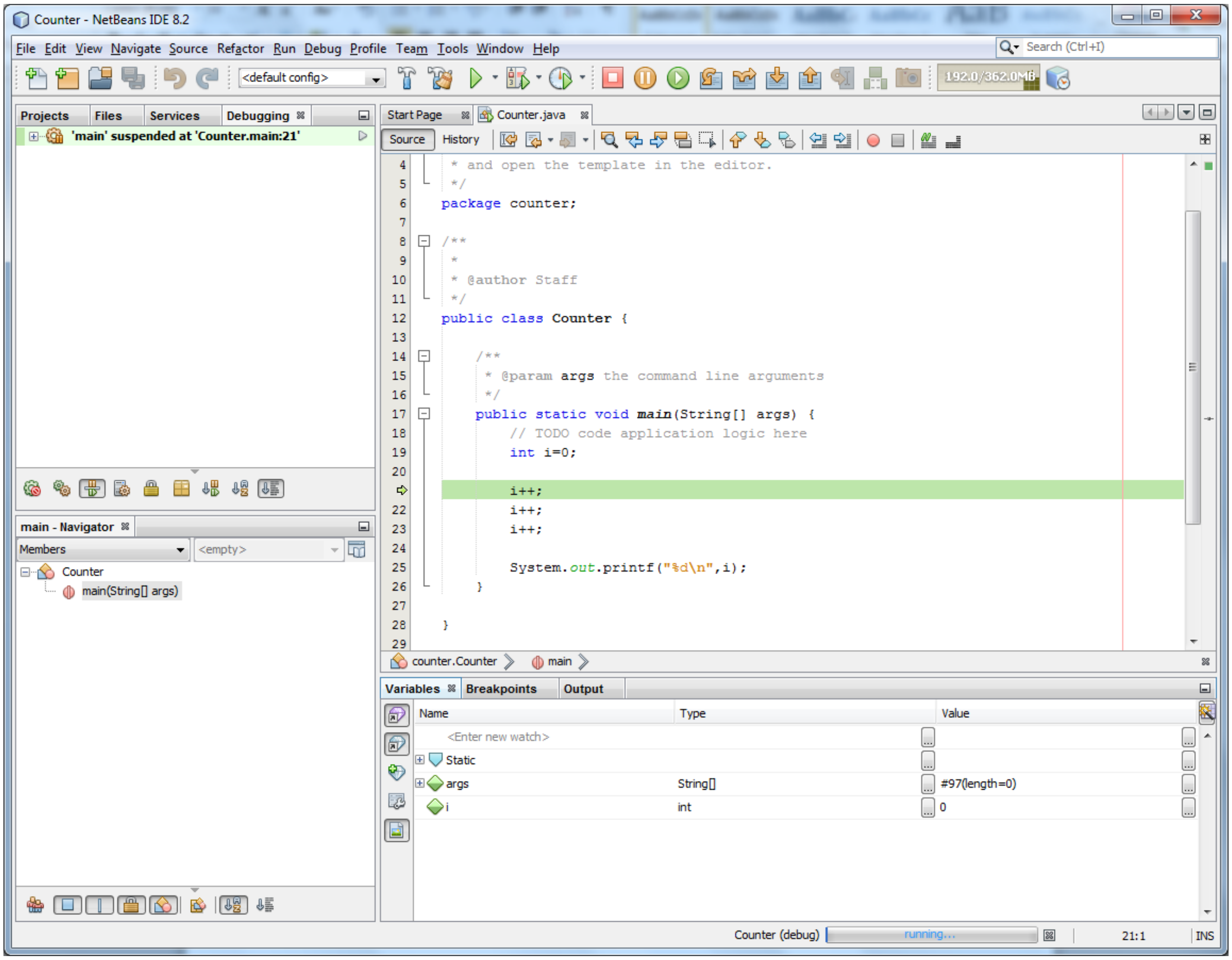
```
19          int i=0;
20
21          i++;
22          i++;
```

In this case we selected in the source code at line 21.



From the **Debug** menu (middle of the menu bar), select **Run to Cursor** (or Press the keyboard shortcut F4).


Debug
Profile
Team
Tools
Windov
Debug Project (Counter) Ctrl+F5
Debug File Ctrl+Shift+F5
Debug Test File Ctrl+Shift+F6
Attach Debugger...
Finish Debugger Session Shift+F5
Pause
Continue F5
Step Over F8
Step Over Expression Shift+F8
Step Into F7
Step Into Next Method Shift+F7
Step Out Ctrl+F7
Run to Cursor F4
Apply Code Changes
Take GUI Snapshot
Set Current Thread...
Stack
Toggle Line Breakpoint Ctrl+F8
New Breakpoint... Ctrl+Shift+F8
New Watch... Ctrl+Shift+F7
Evaluate Expression... Ctrl+F9
Check for Deadlock


Counter - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
192.0/362.0MB
Projects Files Services Debugging
main - Navigator
Members
<empty>
Counter
main(String[] args)
Start Page
Counter.java
Source History
counter.Counter
main
Variables Breakpoints Output

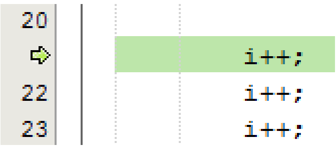
'main' suspended at 'Counter.main:21'

```
 * and open the template in the editor.
 */
package counter;

/**
 *
 * @author Staff
 */
public class Counter {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int i=0;

        i++;
        i++;
        i++;

        System.out.printf("%d\n",i);
    }

}
```

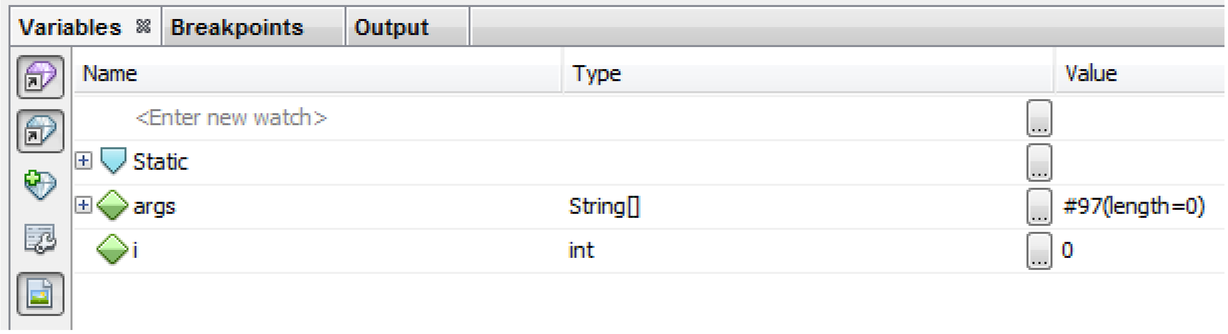
Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#97(length=0)
i	int	0

Counter (debug) running... 21:1 INS

The debuggers stopped the program. We can inspect the variables. Make sure the Variables tab in the lower right in part of the window is selected.

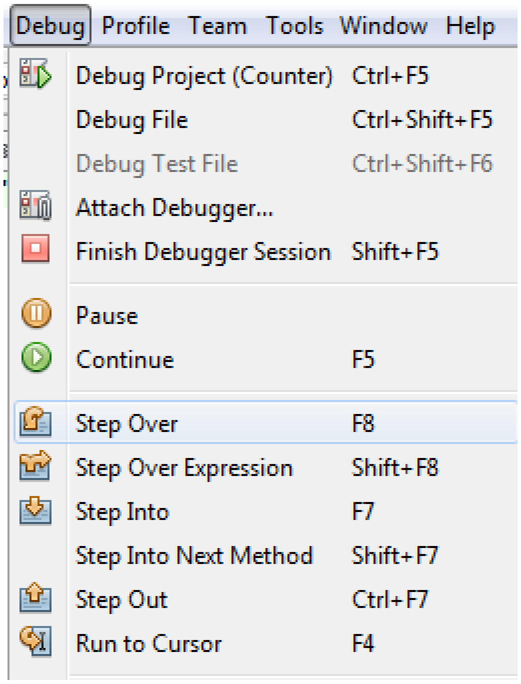


```
        i++;
        i++;
        i++;
```



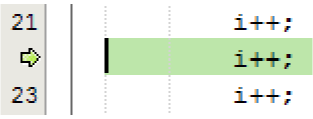
Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#97(length=0)
i	int	0

Notice at the variable i, has the value of 0.

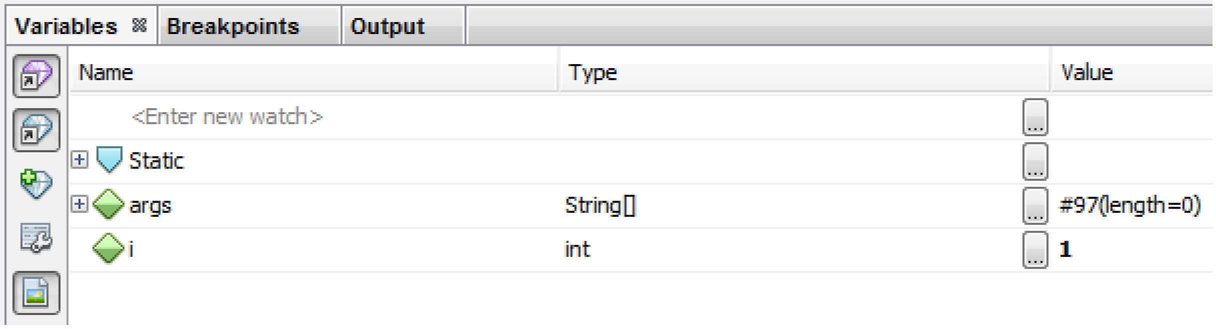

Debug
Profile
Team
Tools
Window
Help
Debug Project (Counter) Ctrl+F5
Debug File Ctrl+Shift+F5
Debug Test File Ctrl+Shift+F6
Attach Debugger...
Finish Debugger Session Shift+F5
Pause
Continue F5
Step Over F8
Step Over Expression Shift+F8
Step Into F7
Step Into Next Method Shift+F7
Step Out Ctrl+F7
Run to Cursor F4

From the Debug menu, select Step Over (easier to press F8). This will step over this instruction to the next instruction.

As you write more complex programs, you may need to Step Into (to see inside of a method).

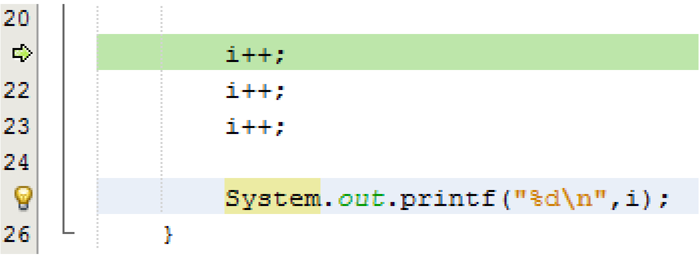


```
        i++;
        i++;
        i++;
```



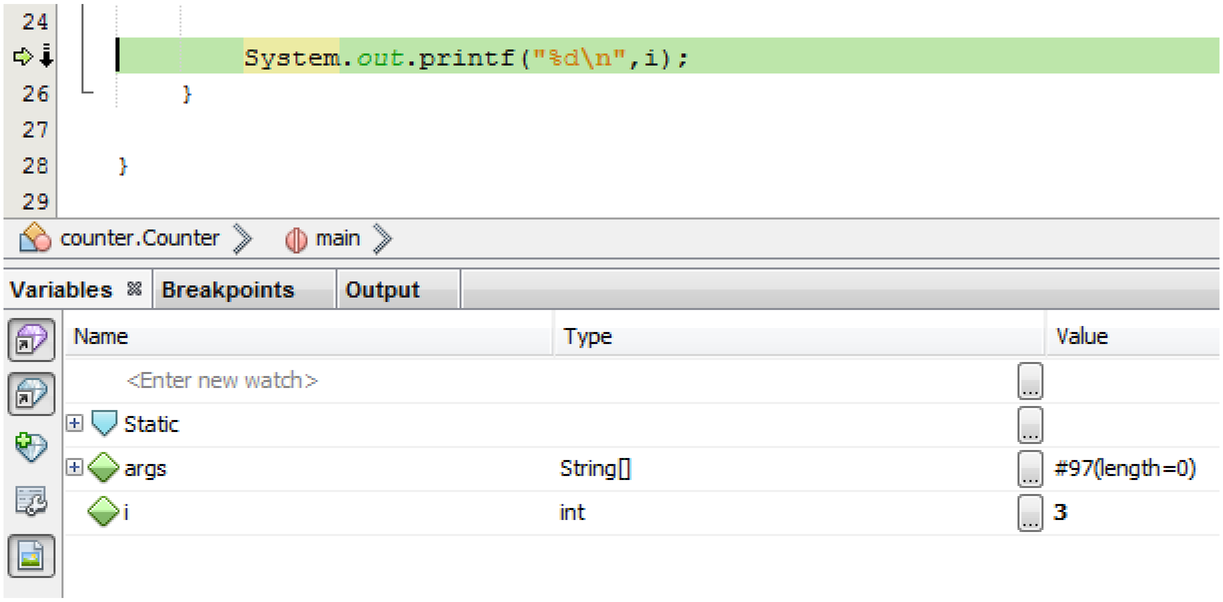
Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#97(length=0)
i	int	**1**

The of i became 1.



```
20
            i++;
22          i++;
23          i++;
24
            System.out.printf("%d\n",i);
26      }
```

If we select line 25 of the program, and select from the debug menu to Run to Cusor (F4).



```
24
            System.out.printf("%d\n",i);
26      }
27
28  }
29
```

Name	Type	Value
<Enter new watch>		
Static		
args	String[]	#97(length=0)
i	int	**3**

It shows that indeed, i has the value of 3, when it will be displayed to the console.