

Lab 00

Eclipse Introduction

Objective:

Learn how to use a basic IDE called, “Eclipse”, and write a very simple program step-by-step.

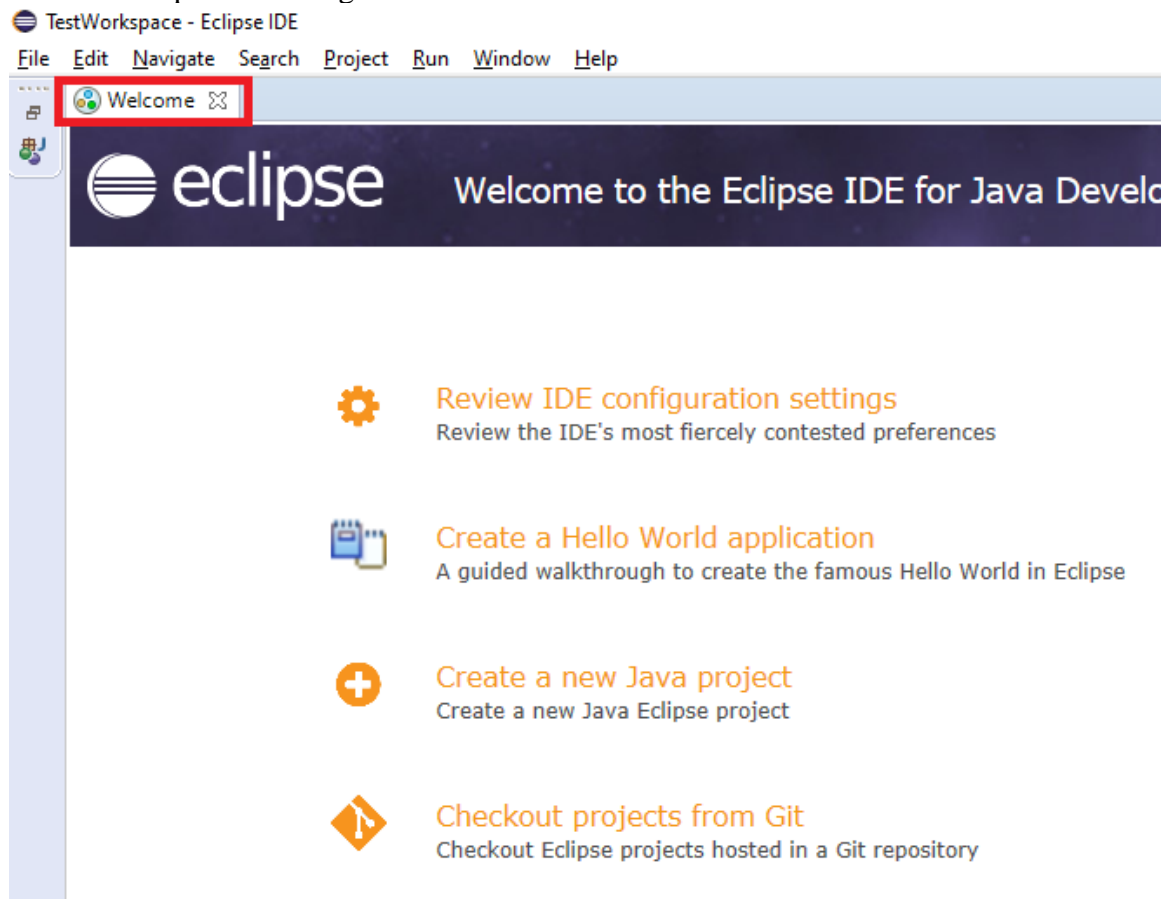
Lab Solution

Requirements:

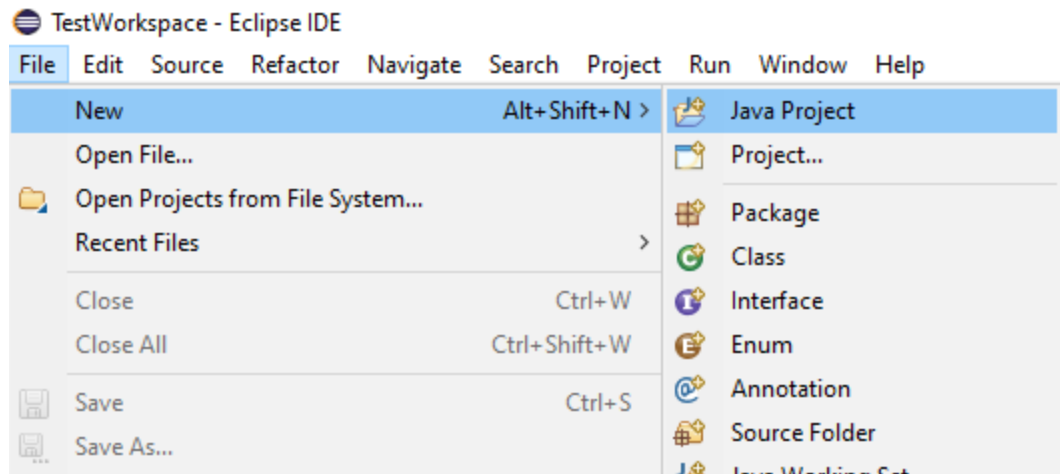
- Follow steps below exactly. (100pts)

Steps:

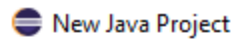
1. Open Eclipse.
2. If there is a splash screen go ahead and close it



3. Create a new project by going to the toolbar and clicking **File->New->Java Project**



4. In the dialog enter the Project's Name as "HelloWorld" and click **Finish**



Create a Java Project

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

5. A little lower in the same dialog box, **Uncheck** the box **Create module-info.java**, and finally select **Finish**.
 - a. NOTE: If the box is accidentally checked, then it may cause syntax errors on the first line of your code. If this happens, the [follow these instructions to fix it](#).

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: HelloWorld

☒ Use default location

Location:

JRE

☒ Use an execution environment JRE:

JavaSE-17

☐ Use a project specific JRE:

jdk-17

☐ Use default JRE 'jdk-17' and workspace compiler preferences

[Configure...](#)

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files

[Configure...](#)

Working sets

☐ Add project to working sets

Working sets:

Module

☐ Create module-info.java file

Module name:

☐ Generate comments

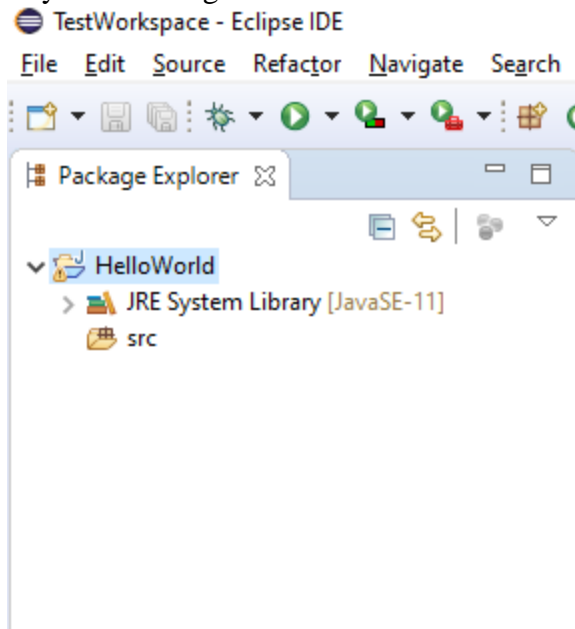


< Back

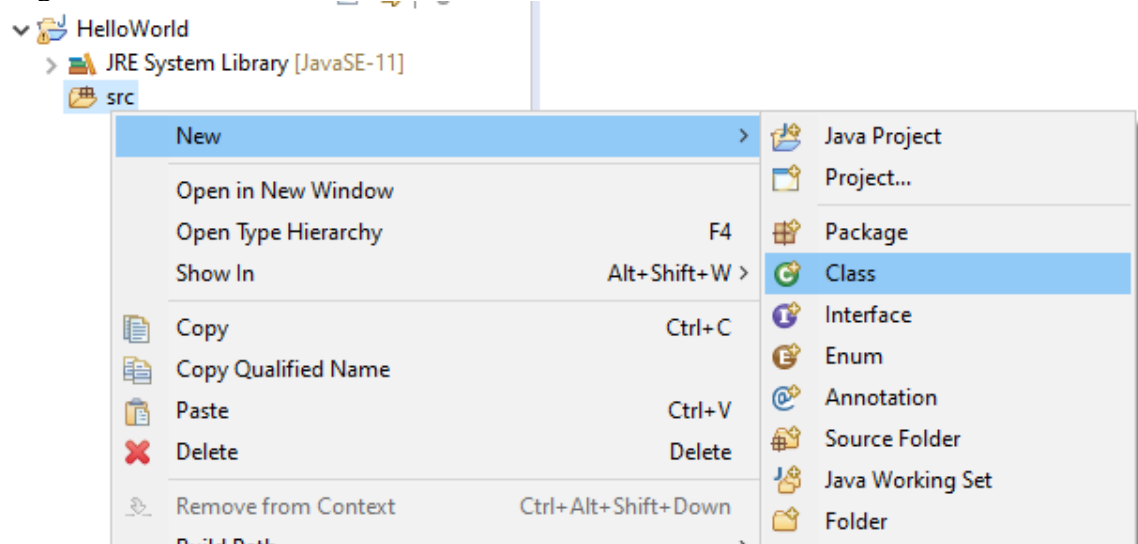
Next >

Finish

6. Now you have a newly created Java project found in the Package Explorer. It is highly recommended that for each assignment you create a new project to keep all of your code organized.




7. Right Click the **src** folder and click on **New->Class**



8. In the next dialog enter the Class name as **HelloWorld**, click the checkbox **public static void main(String[] args)**, and finally click **Finish**

New Java Class

Java Class

 The use of the default package is discouraged.

Source folder: HelloWorld/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: HelloWorld

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...
Remove

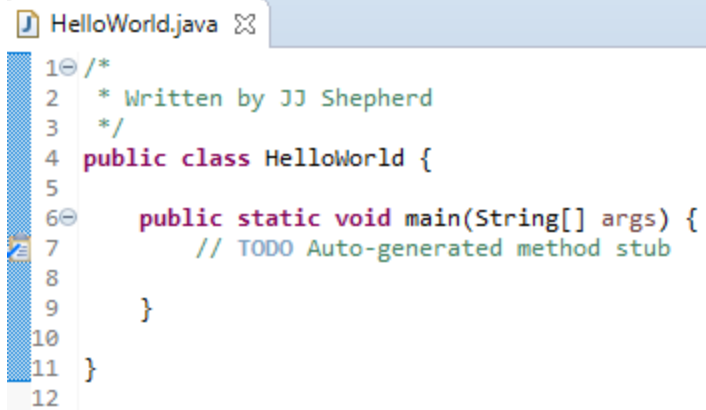
Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Finish Cancel

9. This creates a Java program file. Java code is organized by first “Classes”, and then “Methods”. This creates a Java file whose class is named “HelloWorld” and has the main method. In Java, class names must match the file names or else it will have a syntax error. Most functional code is organized into methods, and the main method is the entry point of a Java program. This is where the computer knows where to start reading the code line-by-line.
10. It should now come up with some code. In the top comments (indicated by /* ... */) make sure to put your name

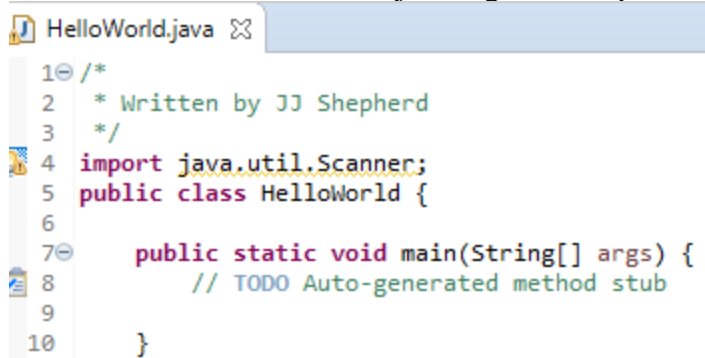


```

1  /*
2   * Written by JJ Shepherd
3   */
4  public class HelloWorld {
5
6      public static void main(String[] args) {
7          // TODO Auto-generated method stub
8
9      }
10
11 }
12

```

11. Next below the comments enter “**import java.util.Scanner;**”. This code will allow us to use the Scanner object to get user input

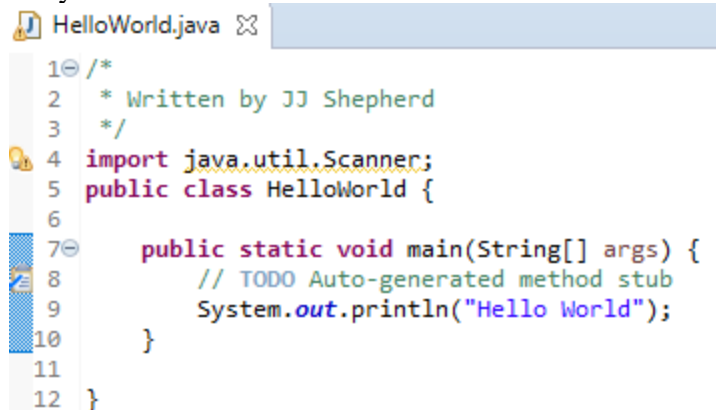


```

1  /*
2   * Written by JJ Shepherd
3   */
4  import java.util.Scanner;
5  public class HelloWorld {
6
7      public static void main(String[] args) {
8          // TODO Auto-generated method stub
9
10     }

```

12. Inside of the main method / entry point (**public static void main(String[] args)**) enter “**System.out.println(“Hello World”);**” This code should be in between the curly braces of the main method.

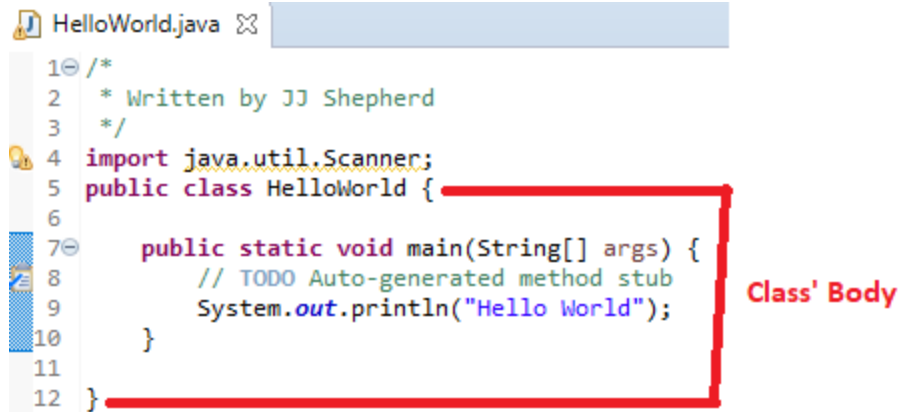


```

1  /*
2   * Written by JJ Shepherd
3   */
4  import java.util.Scanner;
5  public class HelloWorld {
6
7      public static void main(String[] args) {
8          // TODO Auto-generated method stub
9          System.out.println("Hello World");
10     }
11
12 }

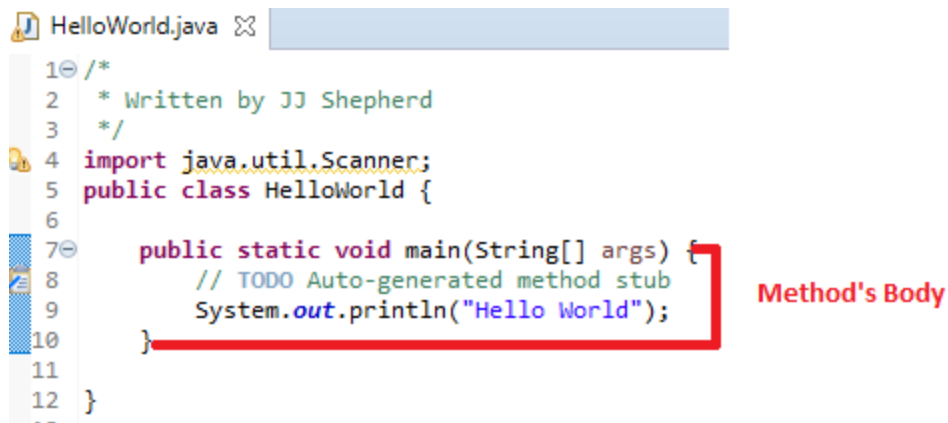
```

13. As stated before, Java code is first organized by Classes and then by Methods. Curly braces ({}) denote the Body of both classes and methods, and must always have a matching pair. Methods are enclosed by the Class’ curly braces, and our line-by-line code is enclosed by the Main Method’s curly braces. For this first part of the class we will only work within the body of the Main Method.



```
1  /*
2   * Written by JJ Shepherd
3   */
4  import java.util.Scanner;
5  public class HelloWorld {
6
7      public static void main(String[] args) {
8          // TODO Auto-generated method stub
9          System.out.println("Hello World");
10     }
11 }
12 }
```

Class' Body



```
1  /*
2   * Written by JJ Shepherd
3   */
4  import java.util.Scanner;
5  public class HelloWorld {
6
7      public static void main(String[] args) {
8          // TODO Auto-generated method stub
9          System.out.println("Hello World");
10     }
11 }
12 }
```

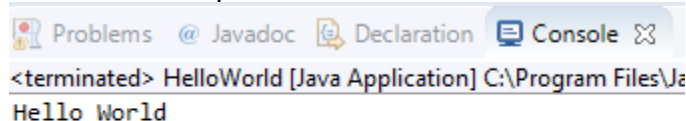
Method's Body

14. Next save the file by **File->Save** and always save frequently

15. Next click on the **Run** button to test your code.



16. This should compile and run the code. If everything is correct, then in the console window it should say "Hello World". The statement "System.out.println();" is a call to the system's standard output. Whatever is put inside of the parenthesis of the statement is printed out to the console.



17. If this is not the case, then check over the code that was entered to make sure there were not any syntax errors. If there were then when you compile, they

should appear in the **Problems** dialog, and will be highlighted in the source code. For instance, if one were to forget to put the semicolon at the end of a statement the error would look like this. Notice the IDE give the line number of where the error occurs, so when errors happen this is the first place to look.

Problems @ Javadoc Declaration Console				
1 error, 1 warning, 0 others				
Description	Resource	Path	Location	Type
✖ Errors (1 item)				
✖ Syntax error, insert ";" to complete BlockStatements	HelloWorld.java...	/HelloWorld/src	line 9	Java Problem
> ⚠ Warnings (1 item)				

```

4 import java.util.Scanner;
5 public class HelloWorld {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         System.out.println("Hello World");
10    }
11
12 }

```

18. Next add the following code after “Hello World” but still inside the main method

```

public static void main(String[] args) {
    // TODO Auto-generated method stub
    System.out.println("Hello World");
    System.out.println("What is your name?");
    Scanner keyboard = new Scanner(System.in);
    String name = keyboard.nextLine();
    System.out.println("Greetings! "+name);
}

```

19. Save it, compile it, and run it. You’ll now notice in the dialog box it waits for you to enter in some information. Enter your name and it should print out “Greetings <YOUR NAME>”.

```

Problems @ Javadoc Declaration Console
<terminated> HelloWorld [Java Application] C:\Program Files\Ja
Hello World
What is your name?
JJ
Greetings! JJ

```

20. The variable “**keyboard**” creates a new instance of an object Scanner. This allows us to gain input from the user via the keyboard. The statement “**String name = keyboard.nextLine();**” creates a variable called “**name**” of type String and then stores the entire line entered in by the user via the method “**nextLine()**”. Finally “**System.out.println("Greetings "+name);**” outputs the salutation plus the contents of the variable “**name**”.
21. Scanner has many methods that can be used to get input from the user. For instance, enter the following code after “**System.out.println("Greetings "+name);**”

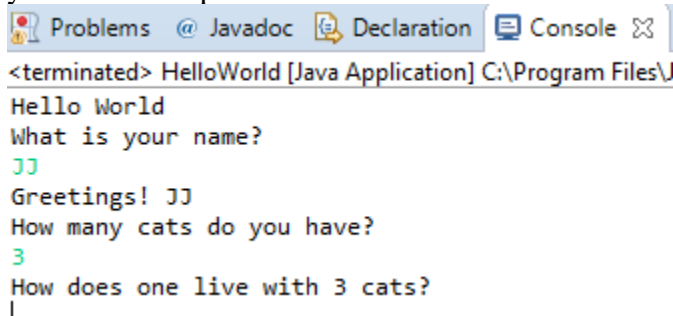

```

public static void main(String[] args) {
    // TODO Auto-generated method stub
    System.out.println("Hello World");
    System.out.println("What is your name?");
    Scanner keyboard = new Scanner(System.in);
    String name = keyboard.nextLine();
    System.out.println("Greetings! "+name);

    System.out.println("How many cats do you have?");
    int numberOfCats = keyboard.nextInt();
    System.out.println("How does one live with "+numberOfCats+" cats?");
}

```

22. Save it, compile it, and run it. Now you enter your name and the number of cats you own. The print out should look similar to the following



```

<terminated> HelloWorld [Java Application] C:\Program Files\J
Hello World
What is your name?
JJ
Greetings! JJ
How many cats do you have?
3
How does one live with 3 cats?
|

```

23. In this example, a new variable of type `int` (integer / whole number) is created and the scanner assigns the numeric value using the method “`nextInt()`”.
24. The contents of “`numberOfCats`” is printed out along with the additional phrase.
25. Congratulations you have finished your first lab! Make sure to submit it to the Dropbox.

Solution Tests:

1. Is your name written as a comment in all source files?
2. Does the solution compile (no syntax errors)?
3. Given the following input sequence < “JJ”,3> does the program output the following?

```

Hello World
What is your name?
JJ
Greetings! JJ
How many cats do you have?
3
How does one live with 3 cats?

```

4. Given another input sequence < “<<YOUR NAME>>”, <<SOME INTEGER>> > does the program output the following? (The information found in the “<<>>” means some value of your choosing. We assume that the values are valid)

```

Hello World
What is your name?
<<Your Name>>
Greetings! <<Your Name>>

```

How many cats do you have?

<<Some Integer>>

How does one live with <<Some Integer>> cats?

Lab Report:

1. Describe how the course grade is calculated. Make sure to include assignment types and their percentages. (10pts)
2. Where are all assignments submitted and where are the grades + feedback for assignments found? HINT: It's the same place, and it's not Blackboard. (10pts)
3. What is the policy regarding late work? (10pts)
4. What is the policy regarding make-up work? (10pts)
5. What is the policy regarding regrade requests? (10pts)
6. True or False. All assignments must be completed individually. (10pts)
7. For programming assignments, what must be submitted to the CSCE Dropbox and what is its file extension? (10pts)
8. What is the name of the program that takes high-level code and transforms it into machine level code? (10pts)
9. Java creates code just before it is fully compiled to machine code. What is this type of code called? (10pts)
10. Name, describe, and give examples of the 3 types of programming errors. (10pts)

Finally:

Upload the source file (.JAVA extension) to the CSCE Dropbox