UNIVERSITY OF VIRGINIA, DEPARTMENT OF COMPUTER SCIENCE

Basic Java 1 - Introduction to Eclipse and Power Function

Nada Basit and Mark Floryan

January 12, 2022

1 SUMMARY

The goal of this homework is to setup your coding environment in eclipse, and to write a very simple program. You will do the following:

1. Download and install eclipse, setup a new project.
2. Implement a simple program that computes the power function.
3. **FILES TO DOWNLOAD:** None
4. **FILE TO SUBMIT:** Power.java

1.1 DOWNLOAD AND INSTALL ECLIPSE

Eclipse is an Integrated Development Environment (IDE) for Java developers. An IDE allows programmers to code in an environment containing more advanced features than a text-editor. Eclipse also provides easy tools for managing projects, compiling code, incorporating outside libraries, etc. Your first step is to download Eclipse. You should be able to find it here.

When you first open Eclipse, you might (depending on your version) see a welcome / tutorial screen (see figure 1.1). If you see this screen, click on **Workbench** in the upper right corner to get to your primary work area. Your workbench, once you find it, should look like figure 1.2.

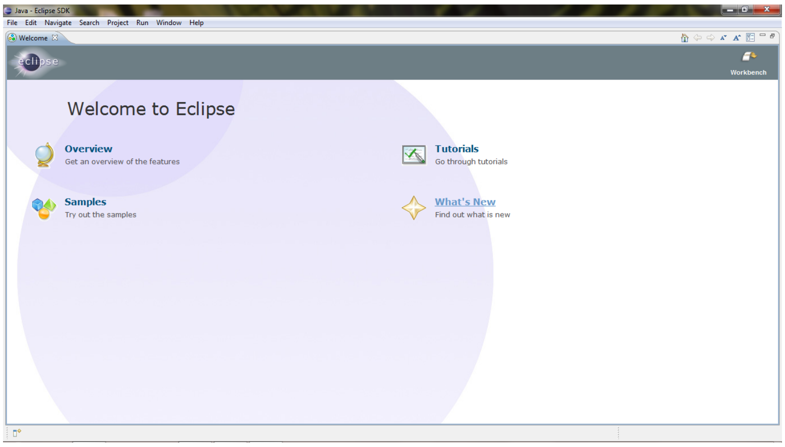
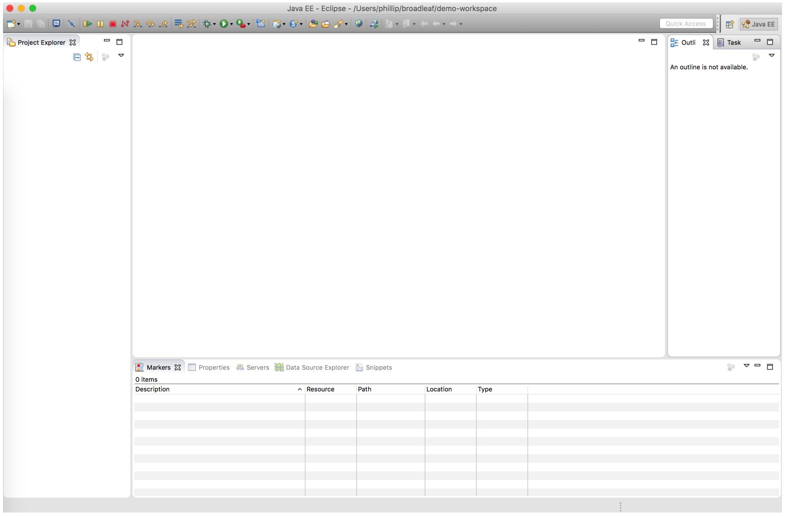


Figure 1.1: Eclipse welcome screen

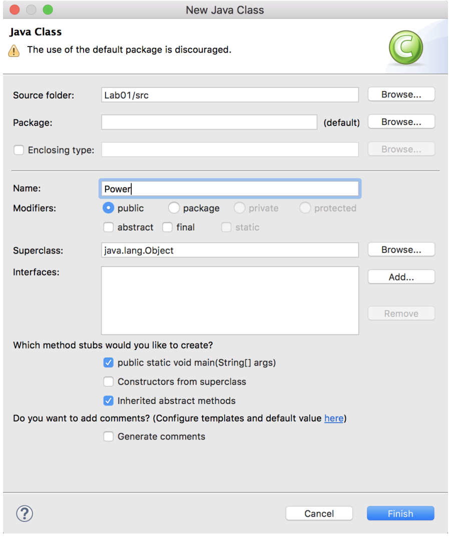


An outline is not available.

Figure 1.2: Eclipse Workspace / Workbench

To start a new project, click on **File –> New –> Java Project**. A dialog will appear. You should give your project a name, select a folder for it (optional), and click finish. You should see your project appear in the left hand dialog of the screen. If you expand the project, you should see a reference to your Java system library, as well as a folder called **src**.

Your next task is create your first Java class. Right click on your project on the left and select **New –> Class**. A dialog will appear (see figure 1.3), name your class **Power**, and select the check box that says **public static void main(String[] args)**. this option will create a main method automatically (though if you forget to do this, you can just type in the main method code manually).



New Java Class

Java Class

⚠ The use of the default package is discouraged.

Source folder: Lab01/src

Package: (default)

□ Enclosing type:

Name: Power

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

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

Figure 1.3: New Class Dialog Screen

You are ready to begin coding. For this homework, you will only need the one class, and you can write all of your code in this area. To run your program, there is a small green bug icon at the top of the screen.

1.2 POWER.JAVA

Now that your environment is setup, you will write one very small Java method as a warm-up. The method signature is:

```java
public static long power(int base, int exp);
```

This method should accept two parameters (base and exp) and return the result of base to the power of exp. For example, *power(2,3)* would return 8 because $2^3 = 8$.

Your Power.java file should also contain a *main method* to handle input and output. Your main method should take in two numbers from the user keyboard. First, the base and second, the exponent. Then, your program should invoke the power method above and present the user (via the console) the value of base to the exponent power. You can assume that when we test your code, all inputs will fit inside of int types, and all correct answers will fit inside of long types. This method may be iterative or recursive, but **may not use any library functions such as Math.pow()**.

SAMPLE INPUT:

```
2
3
```

SAMPLE OUTPUT:

```
8
```

1.3 Gradescope

You should submit your code to *Gradescope*. If you are having trouble with your submission, you should double check the following common problems:

1. Make sure you are only submitting one file, and it is called *Power.java* (exactly, not power.java or pOwer.java).

2. Make sure you remove any *package* statements from your code before submitting. The autograder doesn't expect your file to be inside a package for this assignment.

3. Make sure your output is in the correct format (see above) exactly. You should not be printing ANYTHING else or the autograder will think your output is incorrect.