

# Eclipse Tutorial

We show you how to install and use the IDE (Integrated Development Environment) Eclipse.

## Installation Process

Before you install Eclipse (now version 3.6.1), make sure that you have already installed the Java 5 (or 1.5) platform, Standard Edition (Java SE), which consists of the Java Development Kit (JDK) and the Java Runtime Environment (JRE). If not, follow the link on the Java resources on the course website, and follow the instruction for installing Java.

To download Eclipse, visit website <http://www.eclipse.org/>; click the download button; and choose to download Eclipse SDK 3.6.1. Choose a mirror close to you, and the download window should automatically come up. After the download is finished, you should have a .zip file. Unzip it to the location you want.

You will see a folder named eclipse. Inside the folder, there is an icon that looks like a big blue dot (or Death Star) named eclipse. Double-click it to start Eclipse. You may wish to create a shortcut on the desktop to simplify the launching of Eclipse.

## Launch Eclipse

After you launch Eclipse, it will ask you to specify your workspace, which is a folder in which Eclipse stores all your work for this session. After you specify it, press button OK to continue. Then, you will see some kind of introductory page. Click the curved arrow on the right to go to the main working windows.

## How to create a new project and a new class

Each Java program you write needs to be contained in its own project, which is a folder that contains every class in it. So, we first need to create a new project. To do that, click `File` and choose `New > Java Project`. (Alternately, click the black downward-pointing triangular arrow next to the rectangle in the top left that looks like a mini-window with a blue menu bar and a diamond or star in the upper right corner of it; and select `Java Project` from there. The little diamond is supposed to indicate something shiny and new.) You'll get a dialog box that asks for the Project name; type it in. In this example, we will put `PandaProject` and then click button `Finish`. We have created Java project `PandaProject` (basically a folder in the workspace location).

We are ready to create a new class. There are several ways to do this. You can choose from the menu bar as we did earlier. Click `File` and choose `New > Class`. Another way is to click the same the black downward-pointing triangular arrow [next to the rectangle in the top left that looks like a mini-window with a blue menu bar and a diamond or star in the upper right corner of it] and choose `Class` instead of `Java Project ...`. Also, you can right click the name of the project in the `Package Explorer` tab, then choose `New` then `Class`.

The `New Java Class` window should open. Type in the name of the class—we use `Panda`. At the bottom of the window are several options. One option you may need is method `main`, which is necessary for running the main program. Then, click button `Finish`. Now you will see your class appear below the project name in the `Package Explorer` window.

## How to run a project

Before you try to run a Java program in Eclipse, make sure that you have a method `main` in one of the classes in the project. It is required for running your program. Again, you can obtain `main` method by selecting it when you first created the class or by manually typing `public static void main(String[] args)` as a method header. (Complete the body appropriately, of course).

To run the program, you can click `Run` on the menu bar and choose `Run`. You can also click the icon that is a green circle with a white triangle in the middle. It has a small arrow next to it, so that you can choose `Run`.

If you want to do something fancier, like run the program using some specified arguments, things get a little more complicated. Choose the `Run configurations...` option, which will pull up a tabbed window. However, you need to know only a few major things. On the left side, there is the `Configuration` window, where you can choose to run different types of Java programs. We usually use the `Java Application`. You will see a collection of your classes. Choose the one you want to run (which contains method `main`). Several

## Eclipse Tutorial

tabs will appear on the right. Check at tab `main`. Make sure that the name of the class and project are correct. If not, you can click button `Browse ...` to find the right one. To add arguments, click tab `Arguments`. We are interested in `Program arguments`. Here is where you can put your arguments. For example, if you want the first argument to be "one", the second argument to be "2", and the third one to be "Cornell", put "one 2 Cornell" in the `Program arguments` box.

Now click button `Run`. The result of execution will appear in the `Console` window at the bottom of your main window.

### **Creating a JUnit class**

Creating a `JUnit` class is not as simple as in `DrJava`. First, right click the class for which you want to add `JUnit` test cases. A window will appear asking if you want to add "`junit.jar`" on the build path. Say `Yes`. If you don't know what it is, that is fine. The actual window for creating a new `JUnit Test Case` will appear. The name of your class will be set automatically. In this case, we will put "`PandaTester`". Change the name to whatever you want. You may also want to add a method `main`, which is necessary for running a Java program in Eclipse.

### **Additional Information**

Later, when you develop a large-scale project, you will want to learn more about functionality of Eclipse, such as adding external `.jar` files or debugging tools. Also, Eclipse comes with a lot of plug-ins, for example, for GUIs, and even a `DrJava` plugin. You can find many good tutorials on the internet.