

Project 1

Installing the JDK, Eclipse, and Writing Your First Program

Objectives: Each project handout will begin with a list of objectives you must complete to receive full credit. This list will usually consist of features to include in your code. For this project, however, most of the points are given for creating folders, installing software, and locating documentation. You will also write your first program, but we will provide most of the code. Here are the objectives:

1. Create a folder structure to store files for this class. (10 points)
2. Install the JDK. (20 points)
3. Install Eclipse. (20 points)
4. Write a Java program. (20 points)
5. Compile and run your Java program. (20 points)
6. Locate the String class in the Java API documentation. (10 points)

Submission Instructions: After you complete the project, show a TA (either during the laboratory or during their office hours) that you have done the following:

1. Created the folders described below.
2. Executed your program in Eclipse with no errors.
3. Found the String class documentation.
4. Uploaded a copy of your source code (the .java file) to Zybooks.

Future projects will not require you to have a TA check your work, but we want to make sure everyone knows how to create a simple program in Eclipse before moving on to more sophisticated (and fun!) programs.

Deadline: Ideally, this project should be submitted by Sept. 7 at 11:59 PM. However, because completing the objectives is necessary to complete other assignments, you can submit the project anytime during the semester. Please do not take this as an invitation to procrastinate. You will enjoy the material in this course much more once you can create your own programs.

Recommendation on Backing Up Work: Bad things happen to computers, especially laptops. They break; they get stolen. Lots of hard work can be lost.

One of the more prudent things you should do as a regular computer user is back up your important files. Free services like Dropbox, Google Drive, and Microsoft OneDrive make this relatively painless. I recommend you sync the folders described in the next section to one of these services. This could save you plenty of heartache in the future if tragedy befalls your laptop.

Create Folders: Over the course of this semester, you will accumulate hundreds of files for this class. If you put them all on your desktop, you will have difficulty finding important documents. Your life will be much easier if you organize these files in some sort of folder structure. You

don't have to use the exact structure described below, but I encourage you to use something similar.

1. In a convenient location (e.g., your desktop, user, or documents folder), create a folder with the name "Intro to Programming" or "CS 1324" (or "CS 1323").
2. Inside this folder, create a folder named "Projects". (You might also want to create folders with the names "Lectures" and "Homework" to store lecture slides and homework assignments.)

When you've finished these steps, you should have a folder structure that looks something like the following, where indentation indicates a subfolder:

- Intro to Programming
 - Lectures
 - Homework
 - Projects

Install the JDK: The Java Development Kit (JDK) includes software that runs programs written in the Java programming language. This includes programs you write for this course as well as Eclipse, which you will install in the next section.

1. The JDK installer can be download from the following site:
<https://www.oracle.com/java/technologies/javase-jdk14-downloads.html>
2. Click the "Accept License Agreement" button to unlock the download links.
3. There are two files available for both macOS and Windows. macOS users should download the .dmg file (not the .tar.gz file), and Windows users should download the .exe file (not the .zip file).
4. Windows users: Before running the installer, please uninstall any version of Java currently on your laptop. To do this, click on the start button and then the gear icon. This will open the Windows Settings menu. Click on "Apps," which will open a list of installed programs. Scroll through the list and uninstall any app that starts with the word "Java."
5. Run the installer and **keep the default settings**.
6. You can delete the .dmg or .exe file after step 5 is complete.

Install Eclipse: Eclipse is the most popular integrated development environment (IDE) for Java developers. An IDE is software that includes a code editor along with tools to build and run programs.

1. Eclipse can be downloaded from the following site:
<https://www.eclipse.org/downloads/packages/>. Please do not download the installer at the top of the page. Instead, scroll down a bit to "Eclipse IDE for Java Developers" (not "Enterprise Java Developers"), and download the file corresponding to your operating system.
2. macOS users: Open the .dmg file from step 1, and drag the Eclipse folder to the Applications folder. If you would like to add an Eclipse shortcut to the dock, open the Eclipse folder inside the Application folder, and drag the Eclipse launcher to the dock.
3. Windows users: Right-click the .zip file from step 1 and select "Extract All." The unzipped folder can be moved to any location (e.g., the Intro to Programming folder made earlier or Program Files). If you would like to add an Eclipse shortcut to your start menu or

taskbar, open the Eclipse folder, right-click the eclipse.exe application, and select “Pin to Start” or “Pin to taskbar.”

4. You can delete the .dmg or .zip file after completing step 2 or 3.

Update the System Path on Windows: Try to run Eclipse. If you are a Windows user, you will probably receive an error message that says “A Java Runtime Environment (JRE) or Java Development Kit (JDK) must be available in order to run Eclipse.” This occurs because Eclipse is unable to find the JDK, even though it is installed on your laptop. To fix this, we need to add the location of the JDK to the list of folders that Windows searches. (Previous versions of the JDK installer did this automatically, but the current version seems to omit this helpful step.)

1. Right-click the start button and select “System.”
2. In the window that opens, scroll down to “Related settings” and click on “System info.”
3. A second window will open. Click on “Advanced system settings” from the options on the left.
4. A third window will open. Click on “Environment Variables...” at the bottom.
5. A fourth window will open, which contains two lists: user variables and system variables. In the bottom list (system variables), click on “Path” and then click the “Edit...” button.
6. A fifth window will open. Click the “New” button. This will add a new item to the list. Type the following path, without the quotes: “C:\Program Files\Java\jdk-14.0.2\bin”. Then click the “OK” button.
7. Close out of windows three and four by clicking their respective “OK” buttons.
8. Finally, try running Eclipse again. If all went well, the program should now start. If you receive the same error, please get help from a TA.

Write Your First Program: Now for the fun part. With the JDK and Eclipse installed, we can make our own computer programs!

1. When you run Eclipse for the first time, it will ask you to set a workspace folder. Browse to the Projects subfolder inside the Intro to Programming folder you created earlier. Check the box next to “Use this as the default and do not ask again” and click “Launch.”
2. When Eclipse finishes loading, you will see the welcome screen. Click the button above “Workbench” in the top-right corner.
3. Create a new project by clicking on “File” in the top-left corner, then “New,” and then “Java Project.” A new window will open. Enter “Project 1” as the name (without the quotes), and click the “Next” button.
4. Uncheck the box next to “Create module-info.java file,” and click the “Finish” button. You will never need to create this file for any of your projects, and, in fact, it can cause errors when you try to run your program.
5. Create a new class by right-clicking the Project 1 folder in the Package Explorer, select “New,” and then click “Class.” Make sure the Package field is blank, and name the class “Project1” (again, without quotes). Note that class names should always start with a capital letter and cannot contain spaces. Click the “Finish” button.

Eclipse will add a new .java file to your project and open it for editing. The file will contain the following code:

```
public class Project1 {  
  
}
```

Carefully add code to the file so that it matches the program below. Please do not copy and paste the code from this handout, since typing will help you remember it.

```
/**  
 * This is my first Java program.  
 * @author Your Name  
 * @version 1.0  
 */  
public class Project1  
{  
    // This is the main method. The instructions inside  
    // the braces are executed when the program runs.  
    public static void main(String[] args)  
    {  
        System.out.println("What do you call 8 rabbits? "  
                           + "A rabbyte!");  
    }  
}
```

Build and Run Your Program: Click “Run” in the menu bar at the top of the screen and then “Run” from the drop-down menu to compile and run your program. (Alternatively, click the green circle with the white triangle.) If there are no errors in your code, the program will print your message to the console at the bottom of the screen. Congratulations! You’ve completed a crucial step towards becoming a competent computer programmer.

If your code has errors, don’t despair. Carefully compare it to the code above. Programming is very detailed, so mistakes may be tiny and hard to find at first. Remember that capital and lowercase letters are not interchangeable, nor are punctuation elements (commas, periods, semicolons), nor are grouping elements (parentheses, brackets, braces).

Find the String Class Documentation: The last objective of this project is to access the Java API documentation. The Java API contains classes that can be used in your programs. Over the course of the semester, we will make use of a number of these classes. In order to use an API class without introducing errors into your code, you will need to read its documentation.

Using Google searches, see if you can find the documentation for the String class in the Java API. When you find it, consider bookmarking the page in your browser, so you can quickly reference it in the future. Much of the String documentation will look like gibberish right now, but we will soon learn to extract some useful information.

Upload Code to Zybooks: After you complete each project, you need to upload your source code to your Zybook so it can be graded. If you created the folder structure described earlier, your code is in the folder “Intro to Programming\Projects\Project 1\src”. The code is contained in the file Project1.java. Note that .java files are simply text files that contain Java code. They can be opened with any text editor (e.g., Notepad or TextEdit).

Login to Canvas and go to Assignments -> Projects -> Project 1. Click on the link to take you to Zyante. Drag your .java file to the grey box that says “Drag file here or Choose on hard drive”. This requires a few steps. Click the “Submit Assignment” button in the top-right corner. This will reveal a button near the bottom of the page with the text “Choose File.” Click this button and browse to the location of your .java file. Finally, click the “Submit for grading” button.

The Zybook can only grade 20 points of this assignment. The other 80 points will be graded by the teaching assistant. This will show up as two separate grades in the Canvas gradebook.