

INFSCI 0017 – Fundamentals of Object-Oriented Programming (Fall 2018)

Lab 1

Table of Contents

[Installing and Configuring Java Environment and Development Tools](#)

[Mac OS X](#)

[Windows XP, 7, 8](#)

[Linux](#)

[Installing Eclipse](#)

[Running Eclipse for the first time](#)

[Managing workspaces](#)

[Eclipse IDE \(Integrated Development Environment\)](#)

[Configuring the JRE](#)

[Your first Java program](#)

Topics Reviewed

1. Installing and Configuring Java Environment
2. Working with Eclipse IDE
3. Writing your first Java program

Grading and Submission

You are to write a complete all steps outlined in this document.

Once you have completed all the steps, you should demonstrate your work to your Lab TA.

There will be 5 points for this lab, broken down in the following way:

1. All components of the development environment correctly installed and configured (3 point)
2. Hello World program compiles and runs correctly

Installing and Configuring Java Environment and Development Tools

Note: For this class, we will be using Eclipse IDE (Integrated Development Environment). In order to learn Java and complete assignments for this class, you will need to install the following software packages:

- JRE (Java Runtime Environment)
- JDK (Java Development Kit)
- Eclipse IDE

Mac OS X

If you are running Mac OS X, you already have JRE installed, but you might need to install/update the JDK.

- <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Make sure to select JDK for **MacOS x64**

Java SE Development Kit 8u101		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

Windows XP, 7, 8

- Download and install JRE:
<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>
- Download and install JDK:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Java SE Development Kit 8u101		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

Linux

If you are running Linux, chances are you already know how to install Java on your particular distribution. If not, please schedule an appointment with the instructor or the TA.

Installing Eclipse

You can download and install Eclipse from <https://www.eclipse.org/downloads/>

You do not actually need to install Eclipse – there is no installation wizard. Eclipse is completely self-contained and will run from any location on your computer.

To get started with Eclipse:

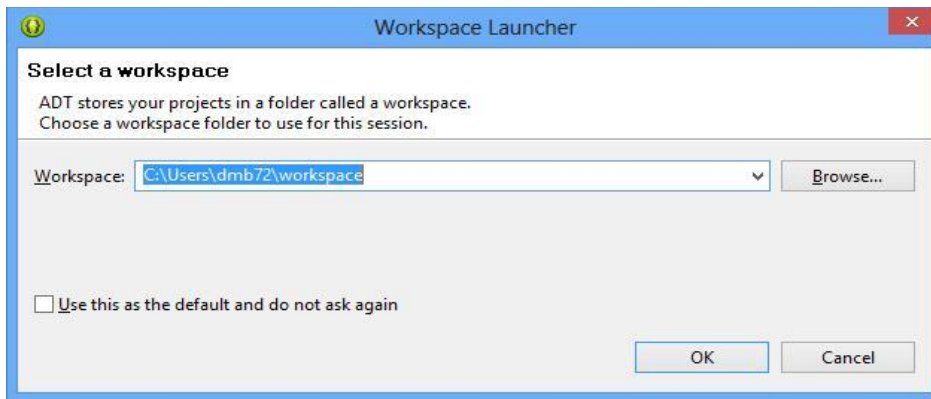
1. Download standard Eclipse from <https://www.eclipse.org/downloads/>. Eclipse website generally does a pretty good job of identifying your computer's operating system, but just in case, make sure that you have download a correct package.
2. Unzip downloaded file
3. If you are a Microsoft Windows user:
 - a. Open Windows Explorer
 - b. Browse to **c:\Program Files**
 - c. Copy unzipped folder (the one you just downloaded) to c:\Program Files\ (you'll end up with c:\Program Files\eclipse)
 - d. In the folder you just copied, locate a file called **eclipse.exe**
 - e. Create a shortcut on your Desktop pointing to **eclipse.exe**
4. If you are a Mac user:
 - a. Open Finder
 - b. Browse to your **Applications** folder (or hit CMND + SHIFT + A)
 - c. Copy unzipped folder (the one you just downloaded) to your **Applications** folder
 - d. Expand the folder you just copied and locate a file called **eclipse.app**

- e. Drag that file to the Dock to create a shortcut. Alternatively, you can run Eclipse by hitting CMND + Space to start Spotlight and start typing the word 'eclipse'.

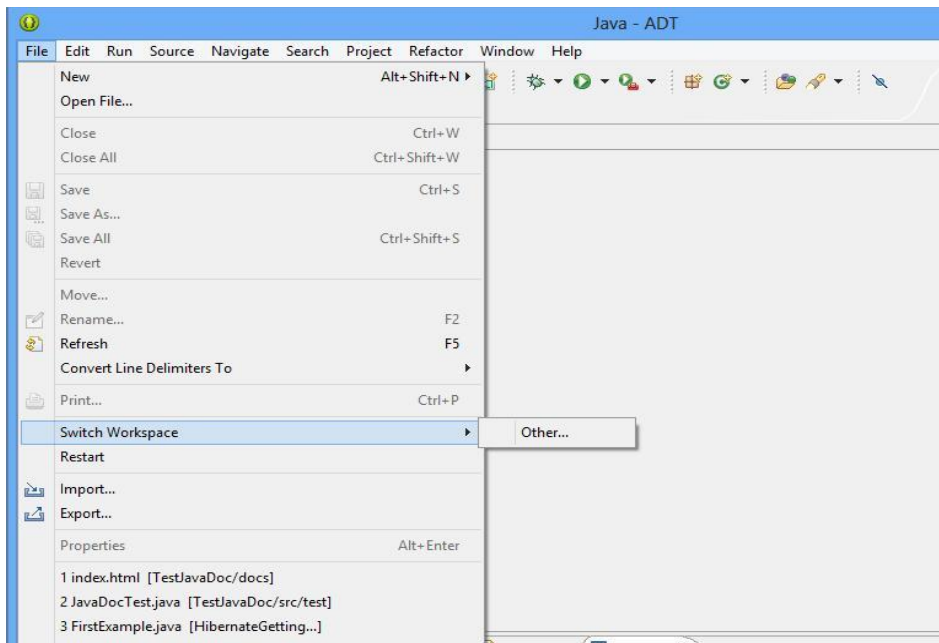
Running Eclipse for the first time

Managing workspaces

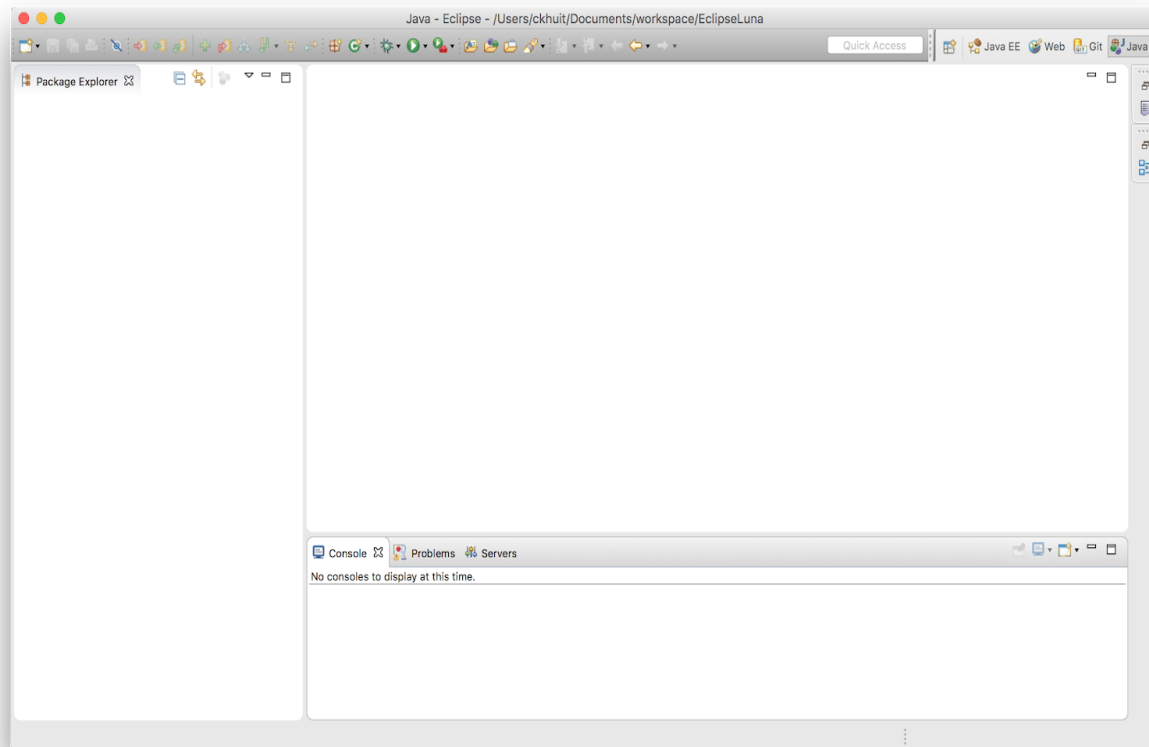
When you start Eclipse, you will see a screen that allows you to select your workspace. A workspace is a folder on your computer where Eclipse saves your Java projects. You can have multiple workspaces depending on what types of projects you are working on. For the purposes of this course, you just need one workspace (the default workspace will work just fine). If you do not want to see this screen pop up every time you launch Eclipse, you can check **Use this as the default and do not ask again** checkbox.

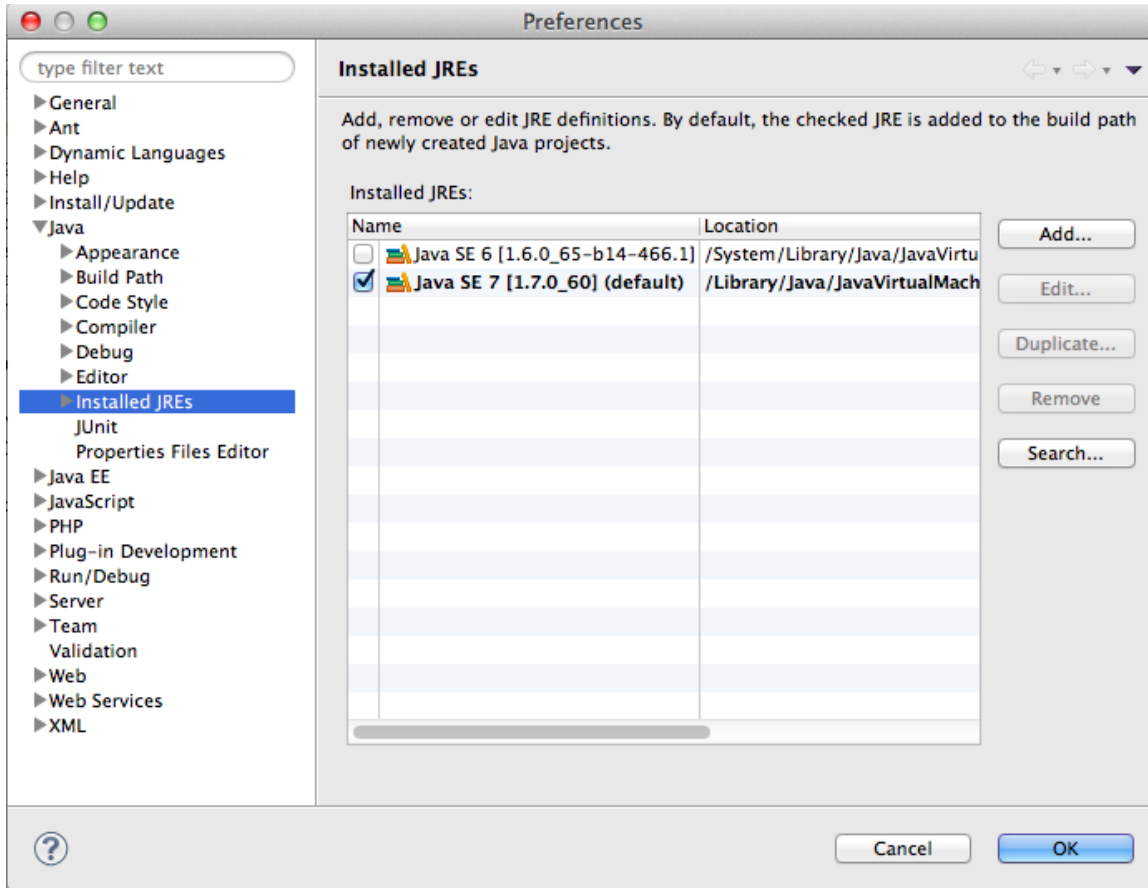


If you need to switch your workspace later on, you can go to File → Switch Workspace.



Eclipse IDE (Integrated Development Environment)





Configuring the JRE

If you are using Mac OS X you will probably need to verify that Eclipse is using the new installed Java.

Note that in these instructions we installed JRE and JDK 1.7. The instructions are basically the same for other versions of JRE or JDK.

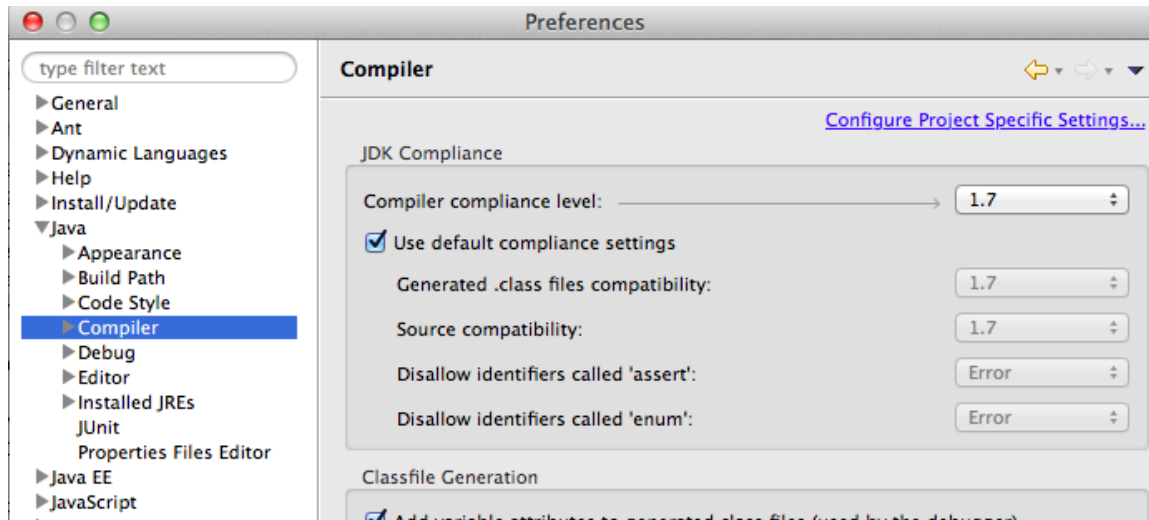
Open the Eclipse preferences (menu Eclipse) and go to Java -> Installed JREs.

(if you install Java 8, consider "1.8" in every place below that put "1.7")

If you can not see the Java 1.7, press "Add", select "Standard VM" and locate where Java 1.7 is. In newer Mac OS X, Java 1.7 is installed in

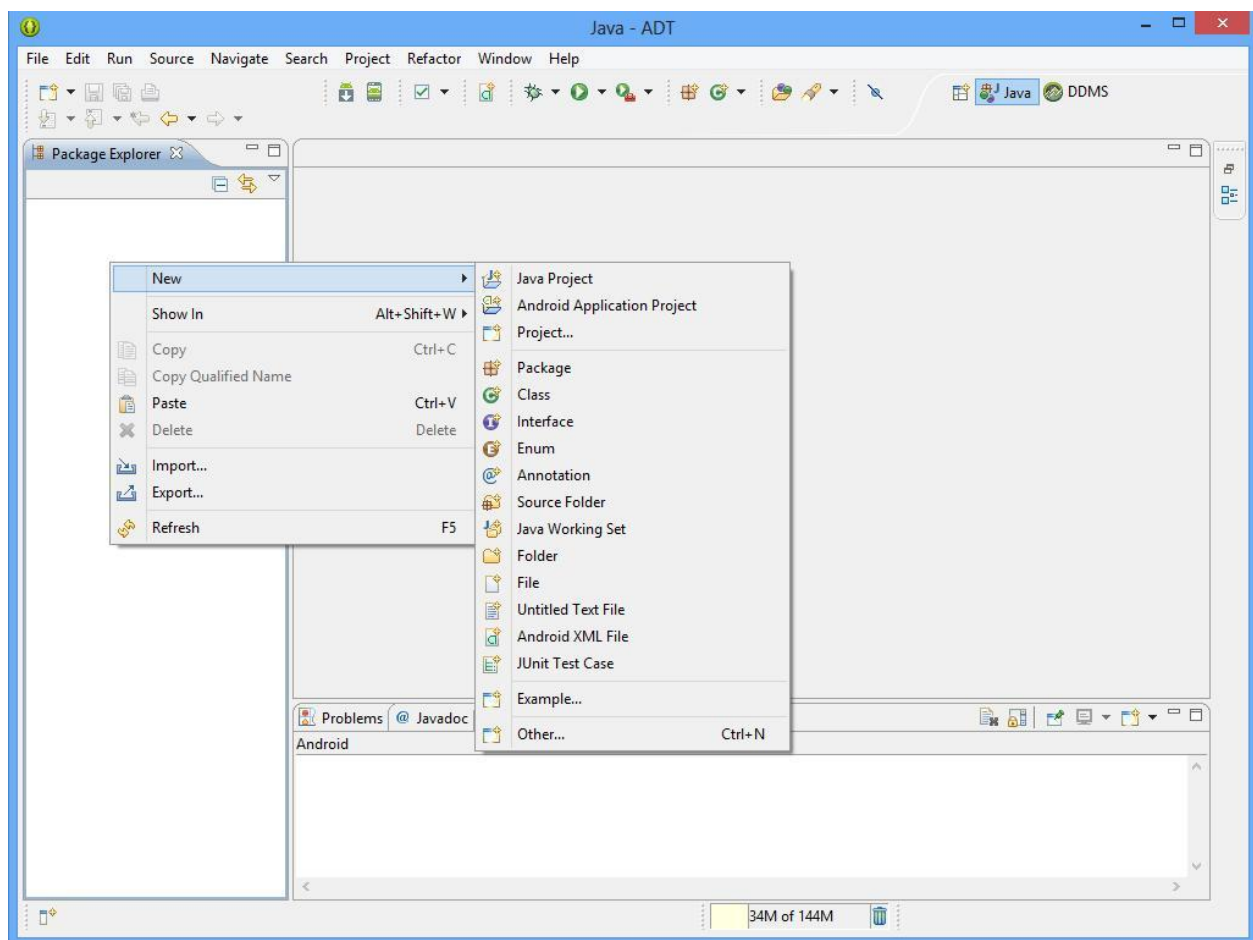
/Library/Java/JavaVirtualMachines/jdk1.7.0_60.jdk/Contents/Home

After adding it, check that compiling version is set to 1.7 too: in preferences, go to Java -> Compiler and select 1.7 from the drop-down.





Your first Java program

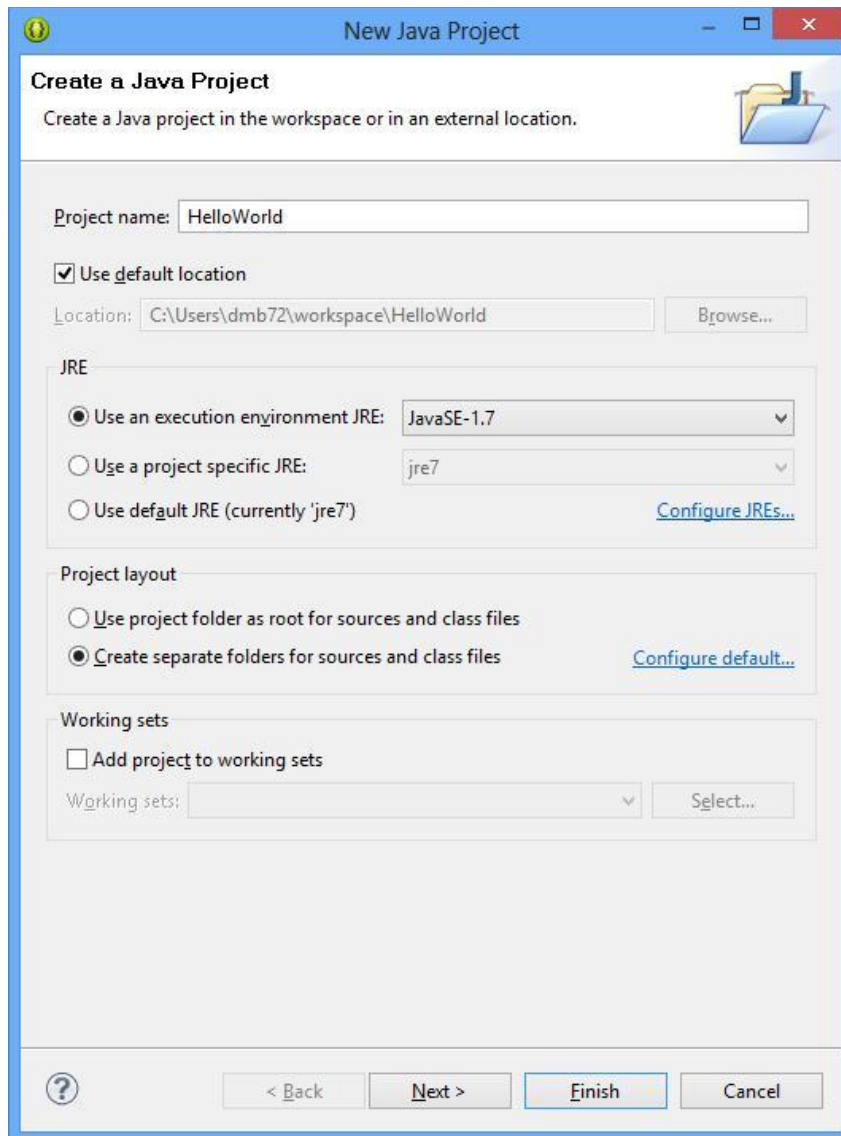
Creating new Java project



There are the four ways to create a Java project:

1. Select **File** → **New** → **Project**
2. Select the arrow  of the button in the upper left of the toolbar. Select **Project**
3. Right click on the Package Explorer view and select **Project**.
4. Click on the  icon on the toolbar.

Select **Java project** and click **Next**.

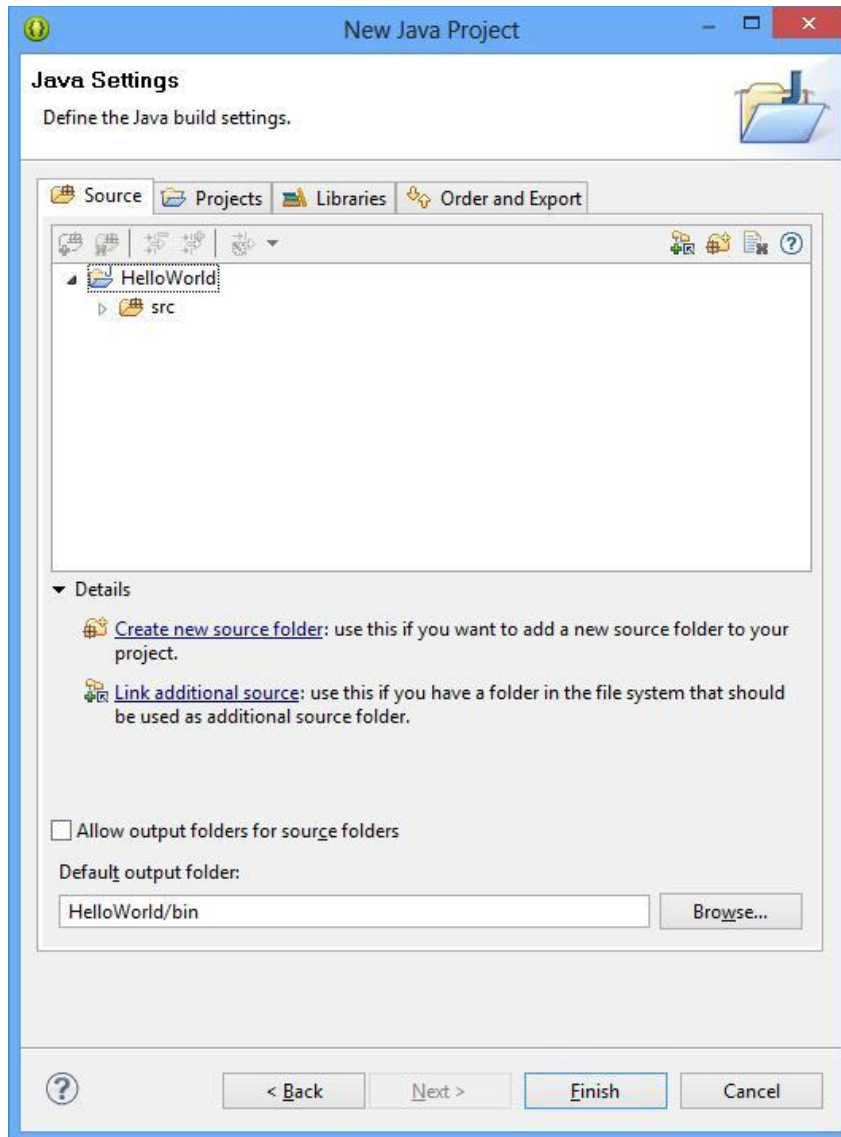


Give the project a name. The accepted naming convention for Java projects dictates that project names follow the rules listed below:

1. No spaces in project names

2. If your project name has more than one word, each word must be (ex: HelloWorld, MusicRecorder, FileReader, etc...). This type of naming/notation is often called “CamelCase” or “CamelBack” notation (<http://en.wikipedia.org/wiki/CamelCase>).
3. For this class, all of your project names must begin with your University of Pittsburgh id (the first part of your Pitt email). For example, if your Pitt ID is “abc123”, and your project name is “Hello World”, you must to name your Java project “abc123_HelloWorld”.one

For now, you can leave all the other fields on this screen at their default values.

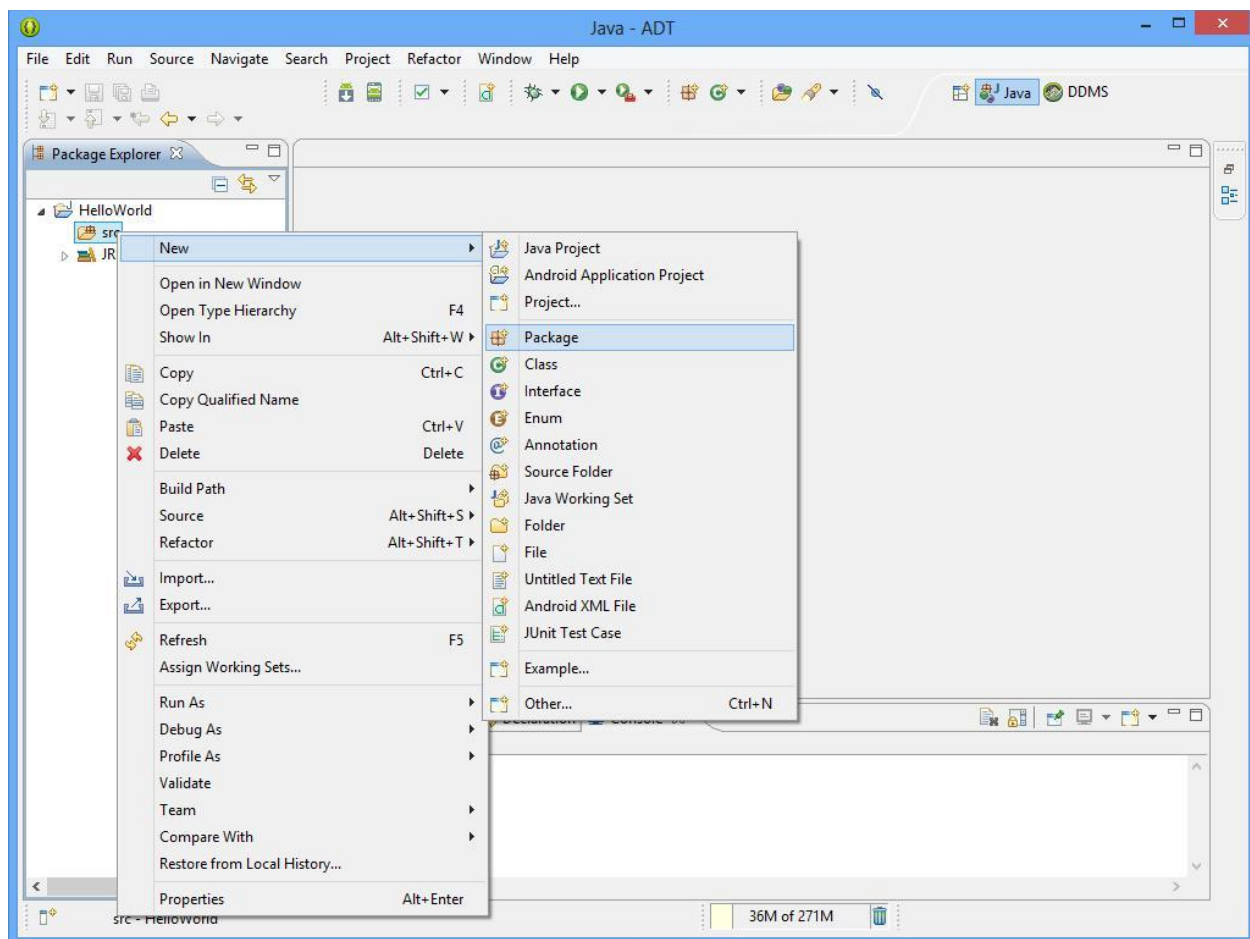


The next screen has four tabs: **Source**, **Projects**, **Libraries**, **Order and Export**.


1. The source tab sets the source and output folders for Java files and class files.
2. The project tab allows you to connect your new Java project to any other Java projects that are needed on the new Java project's build path. 8

3. The libraries tab allows you to attach jar files of and Java libraries that you may need.
 4. The order and export tab determines the build order and how a project can be exported.
- For now, you can leave all the fields on this screen at their default values.

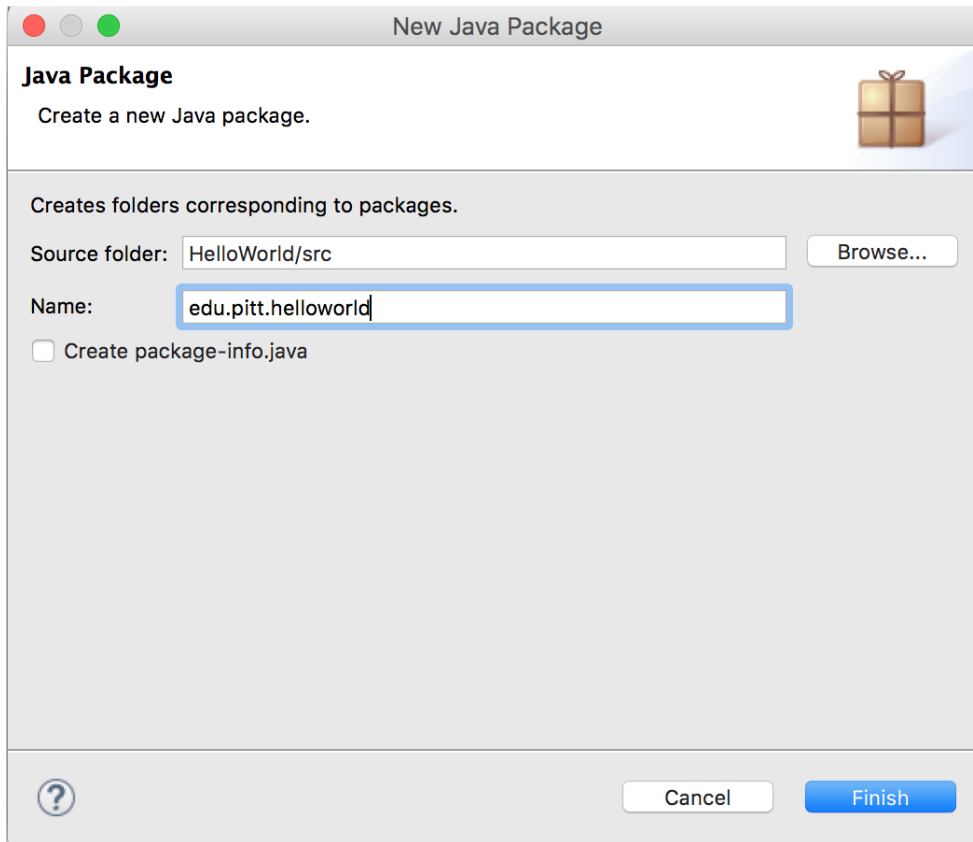
Create new package within your Java project



There are four ways to create a Java Package in Eclipse. First, select the project you wish to create a package in (click on the project name in the Package Explorer).

1. Select **File > New > Other... > Java > Package**
2. Select the arrow  of the button in the upper left of the toolbar. Select **Other... > Java > Package**
3. Right click on a project in the Package Explorer view and select **Package**

4. Click on the  icon in the toolbar while in the Java Perspective.



New Java Package

Java Package

Create a new Java package.

Creates folders corresponding to packages.

Source folder: HelloWorld/src Browse...

Name: edu.pitt.helloworld

☐ Create package-info.java

? Cancel Finish

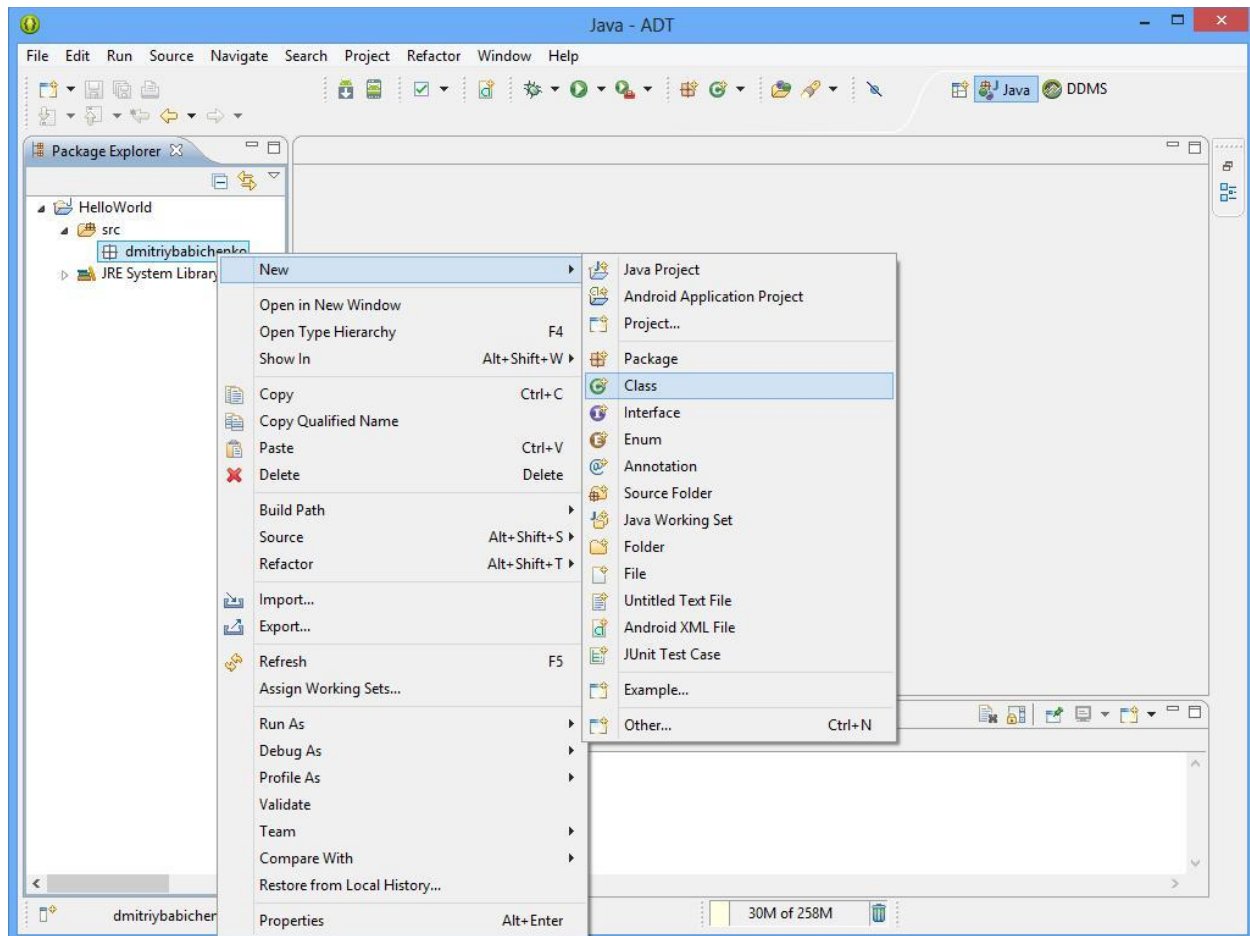
Check to make sure that you are creating the package in the proper project and source folder.

Give the package a name. Make sure to use the following naming conventions for your packages:



1. Package names cannot contain spaces, underscores or any other non-alphanumeric characters.
2. Package names are written in **all lower case** to avoid conflict with the names of classes or interfaces. (We'll learn about classes and interfaces later in the semester)
3. Companies use their reversed Internet domain name to begin their package names. For example, if you are creating a Java package for a project for the University of Pittsburgh, you would name your package "edu.pitt.mypackage" for a package named mypackage.
4. For this class, all of your package names must begin with "edu.pitt." followed by the name of your package.
5. For this example, name your package "edu.pitt.helloworld"

Click **Finish**.

Creating new Java class



There are four ways to create a Java Class. First, select the package you wish to create the class in (click on the package name in Eclipse Package Explorer).

1. Select **File > New > Other... > Java > Class**
2. Select the arrow  of the button in the upper left of the toolbar. Select **Other... > Java > Class**
3. Right-click on a package in the Package Explorer view and select **Class**
4. Click on the arrow of the  icon in the toolbar and select **Class**.

here)' with a 'Generate comments' checkbox (unchecked). At the bottom, there is a help icon (question mark in a circle), a 'Cancel' button, and a 'Finish' button."/>

New Java Class

Java Class
Create a new Java class.

Source folder: HelloWorld/src Browse...

Package: edu.pitt.helloworld 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

? Cancel Finish

Check to make sure that you are creating the Class in the proper package. Give the class a name.

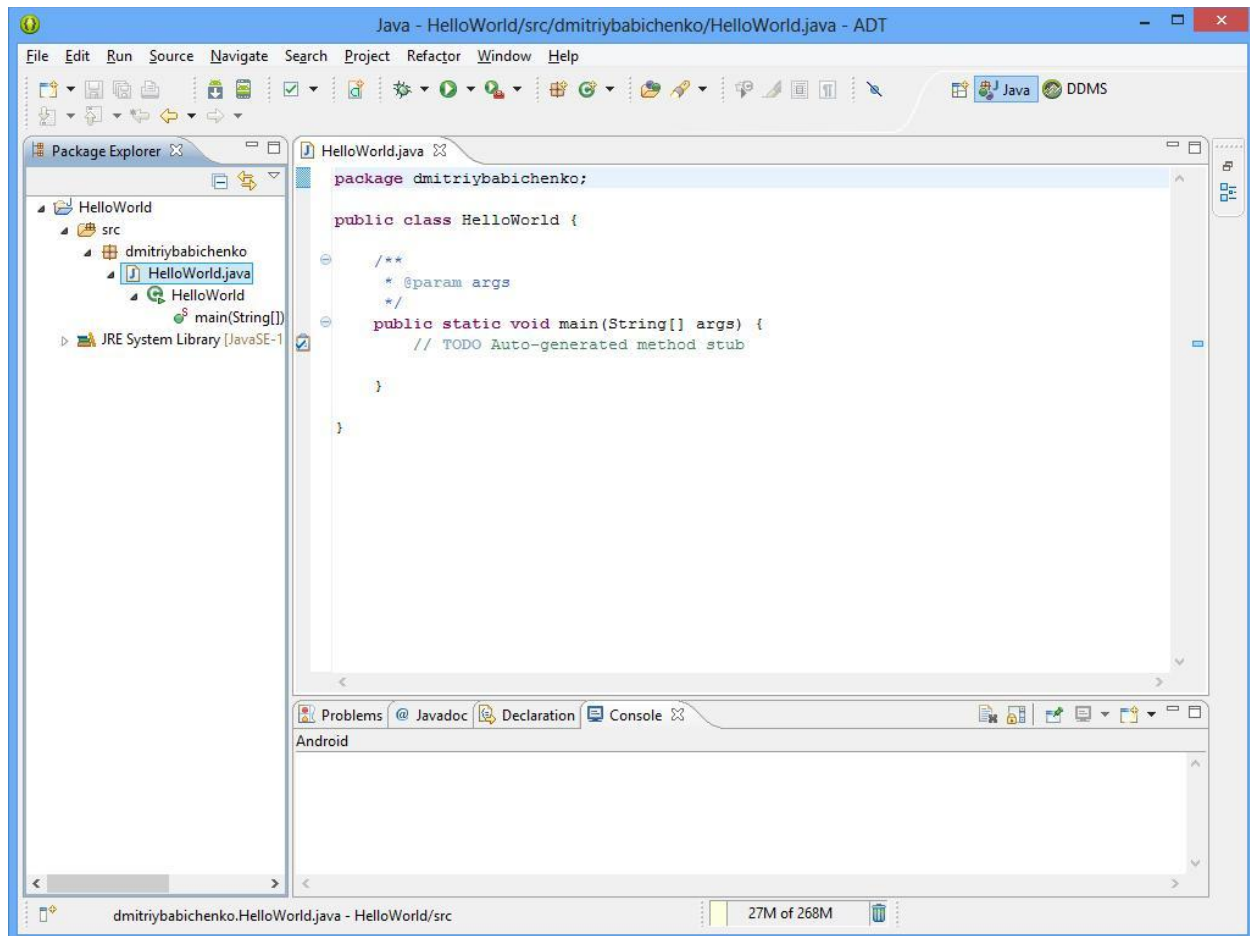
The accepted naming convention for Java classes dictates that class names follow the rules listed below:

1. No spaces in class names.
2. If your project name has more than one word, each word must be capitalized (ex: HelloWorld, MusicRecorder, FileReader, etc...).
3. For this course, do not use any special characters in class names, only words and numbers.

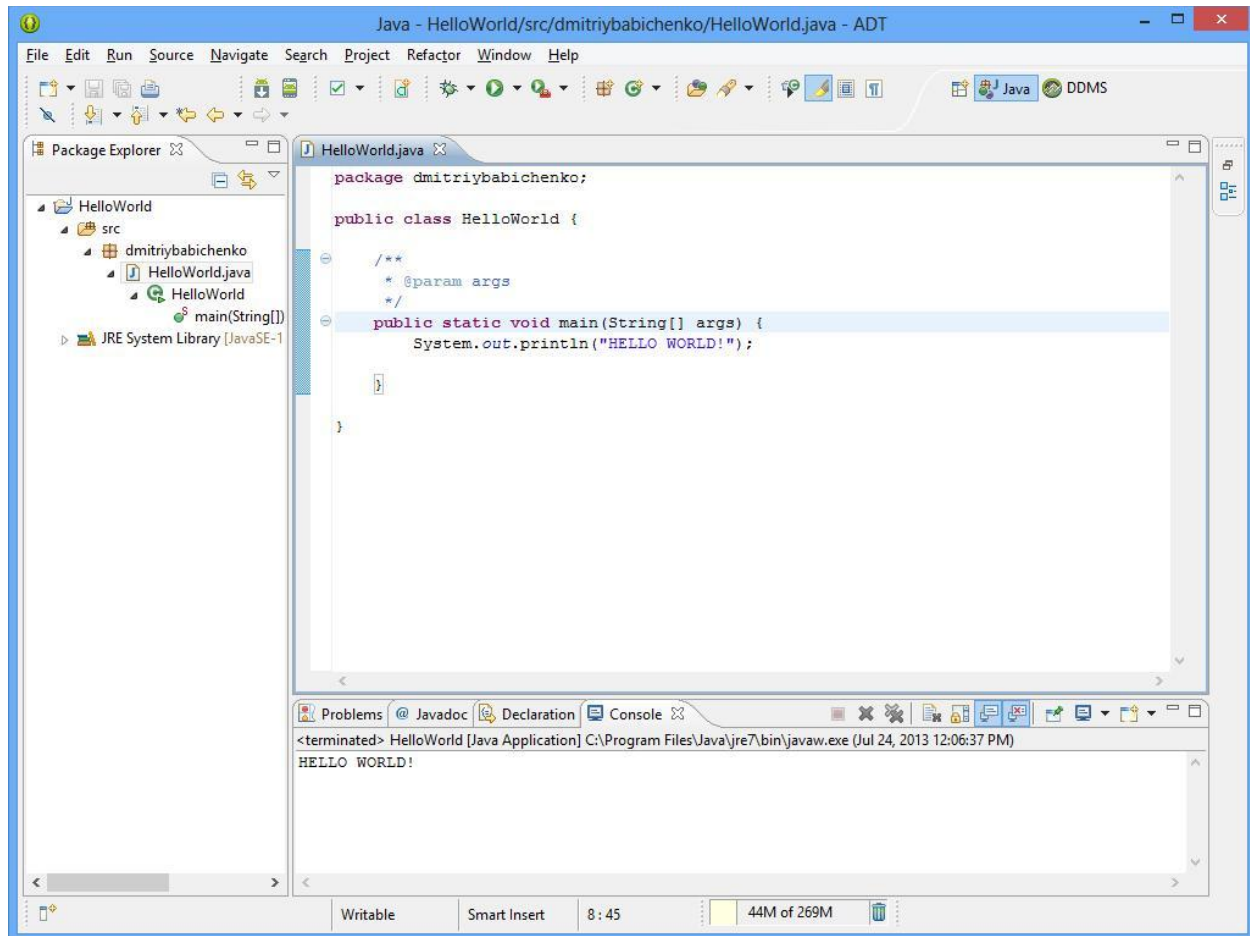
Check the **public static void main(String [] args)** checkbox if the class you are creating starts the application.

For now, you can leave all the other fields on this screen at their default values.

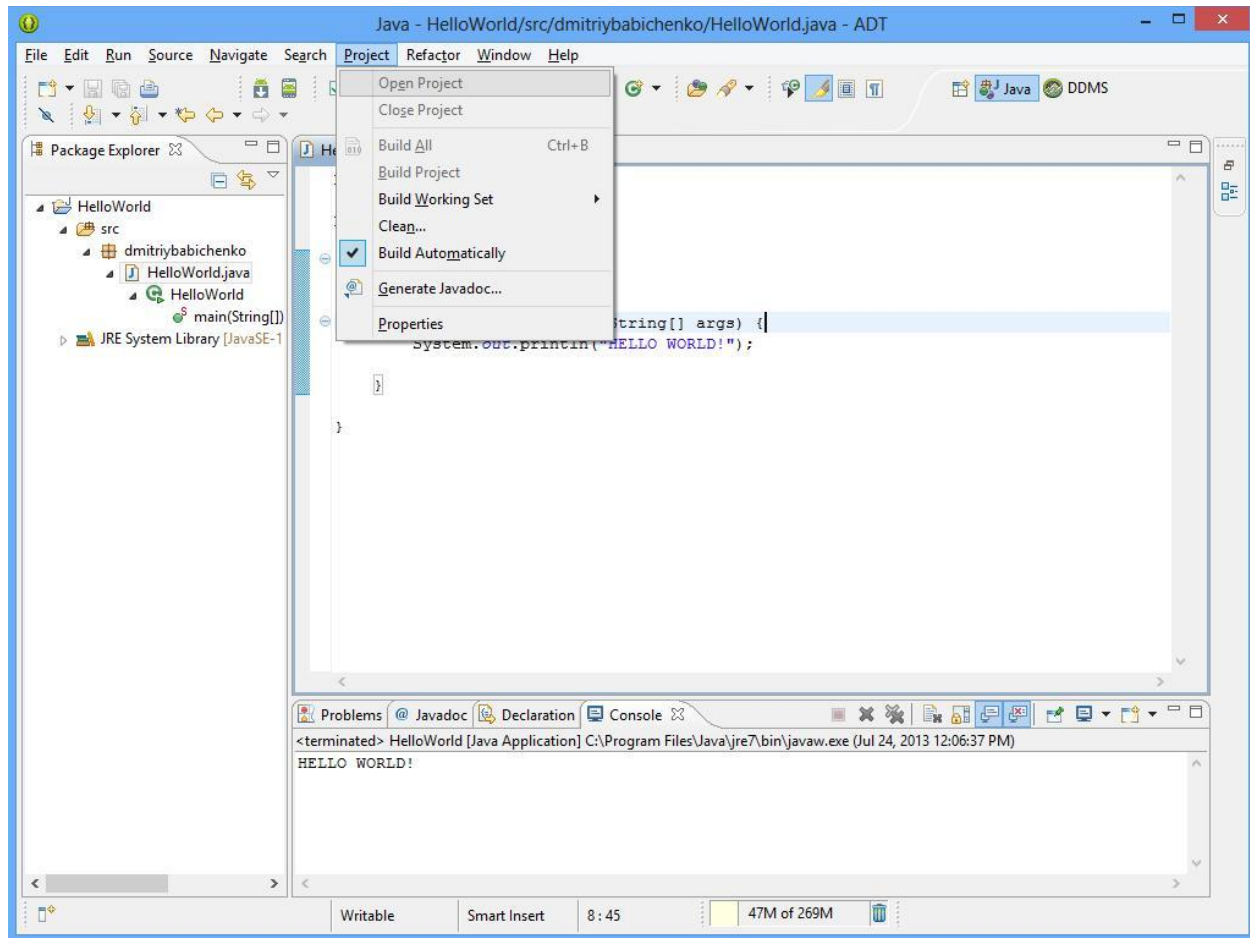
Click **Finish**. The new class will be open in the editor.



Your first “Hello World” program




Running a Java program

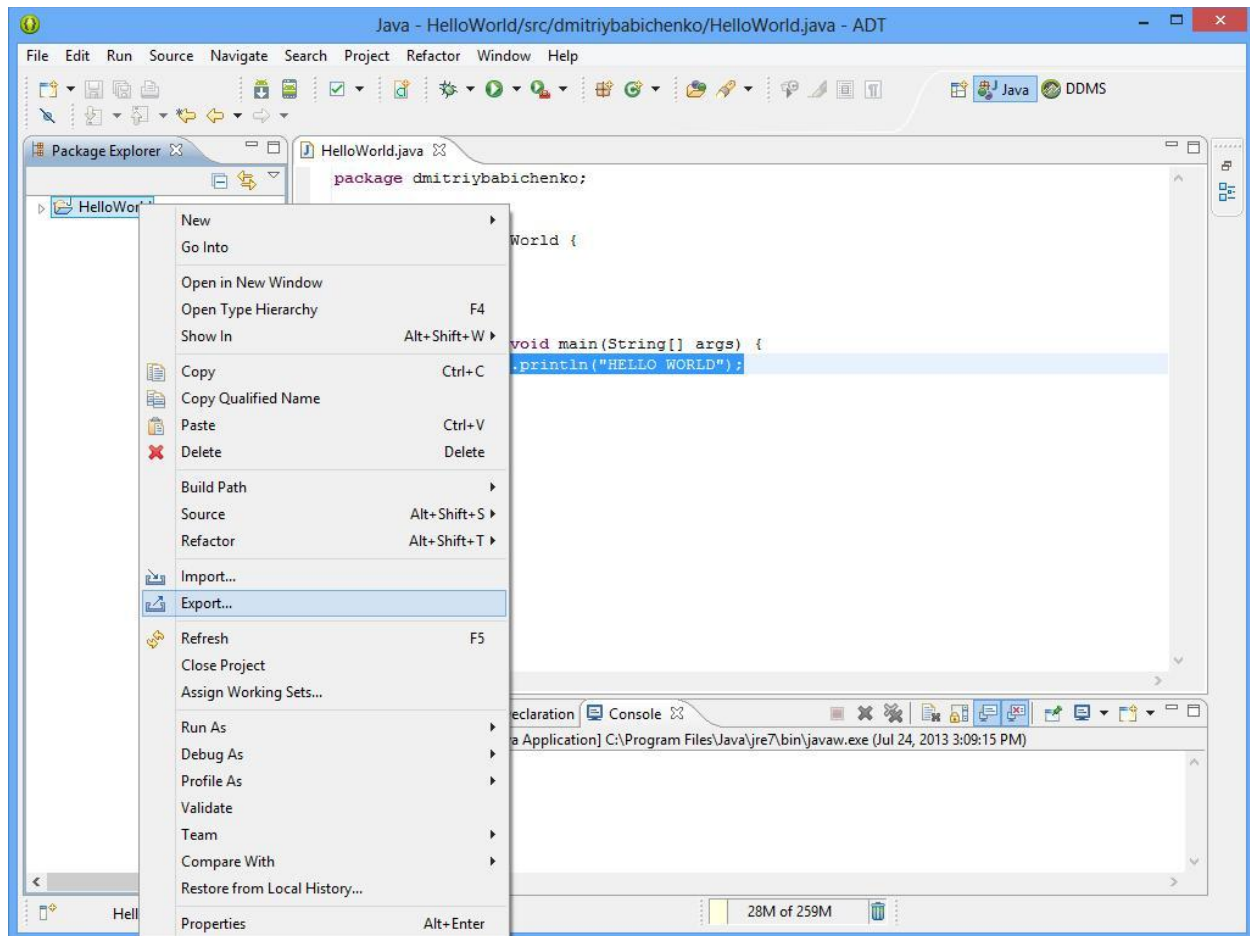


Before you run your program, go to **Project** menu option and make sure that **Build Automatically** is checked (see screenshot above).

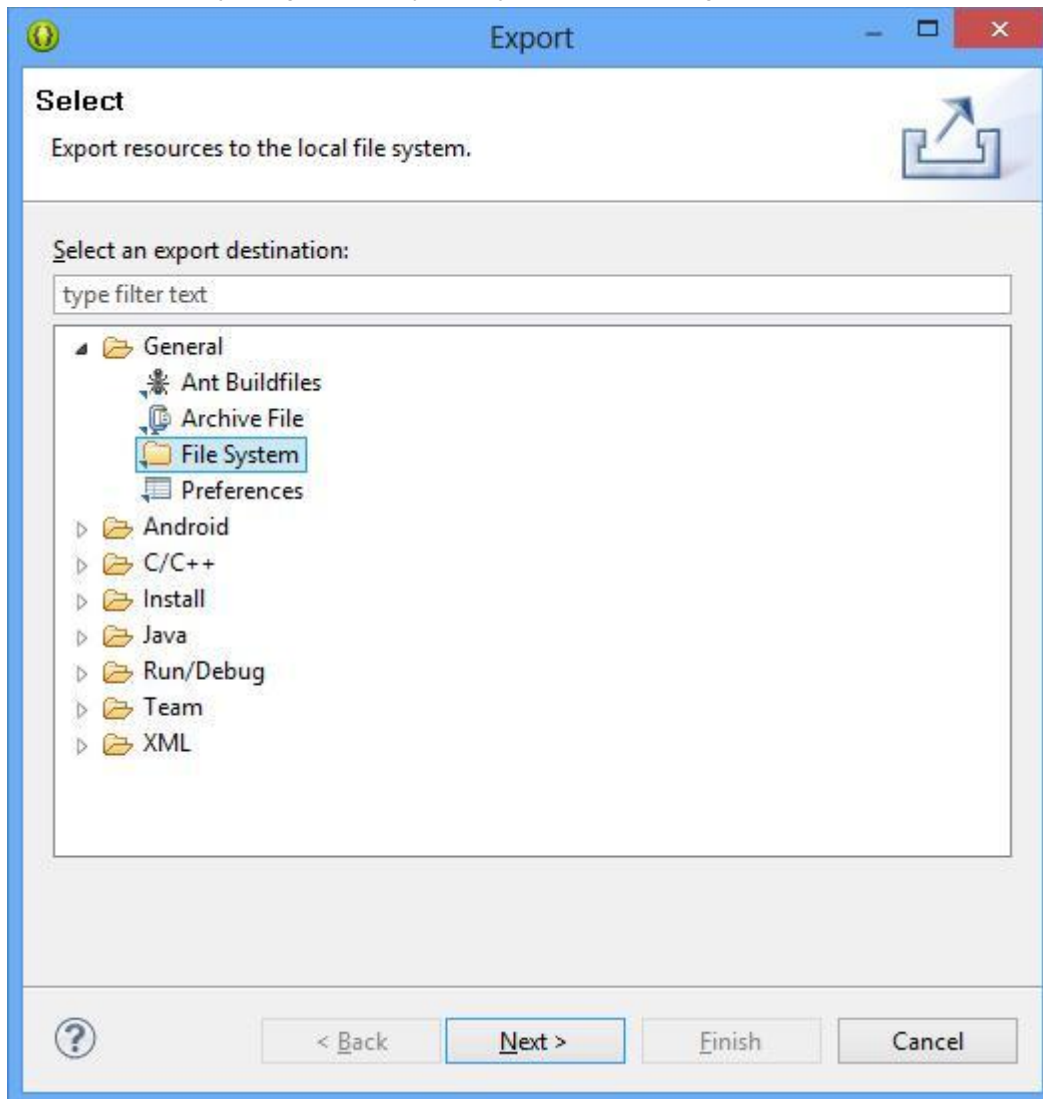
There are 3 ways to run your program from Eclipse:

1. Select **Run > Run** from the menu
2. Click the arrow on the  icon in the tool bar and select **Run**
3. Right click on the project you wish to run. Select **Run > Java Application**

Exporting a Java project



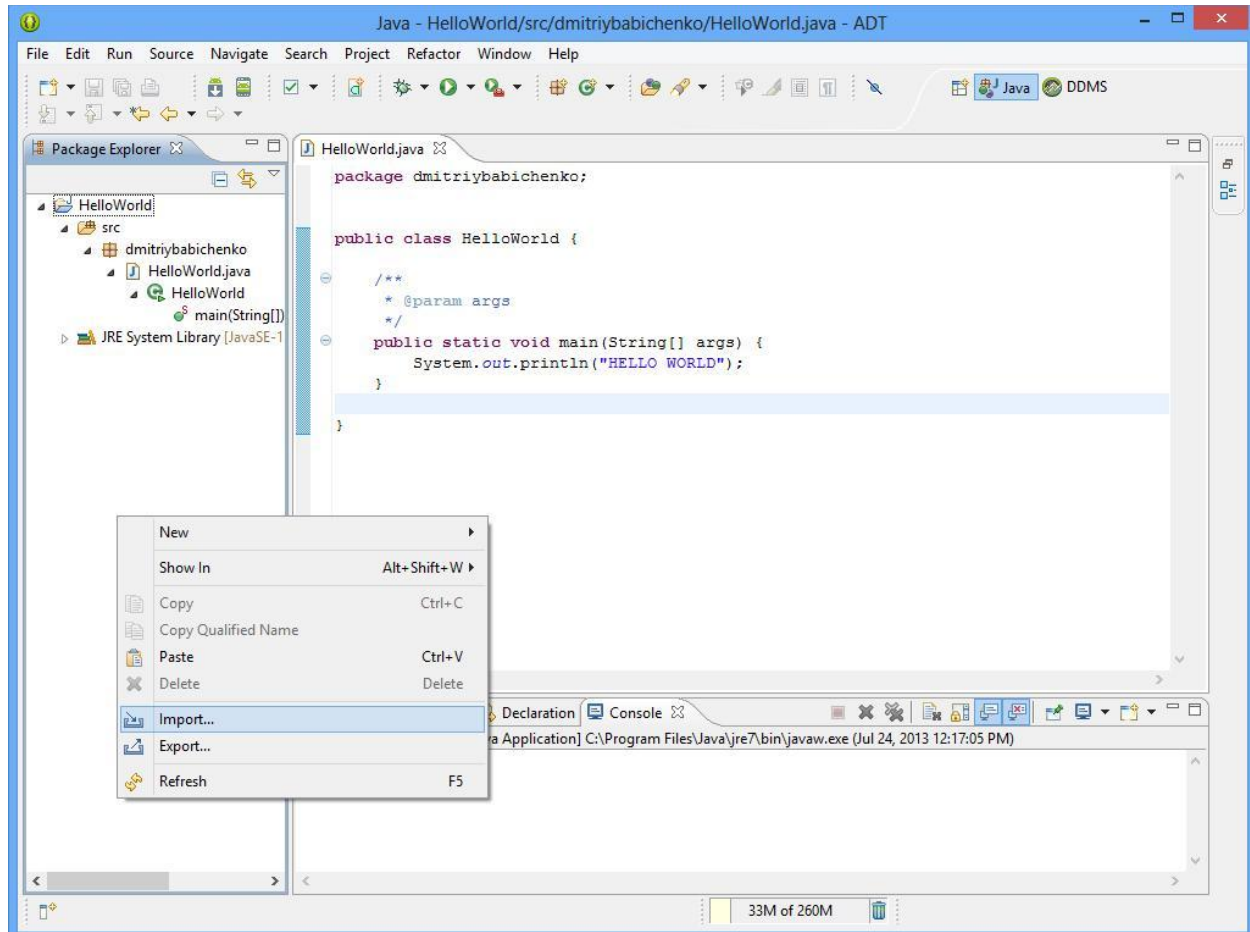
1. To export a Java project, right-click on your project name in Package Explorer and select **Export**



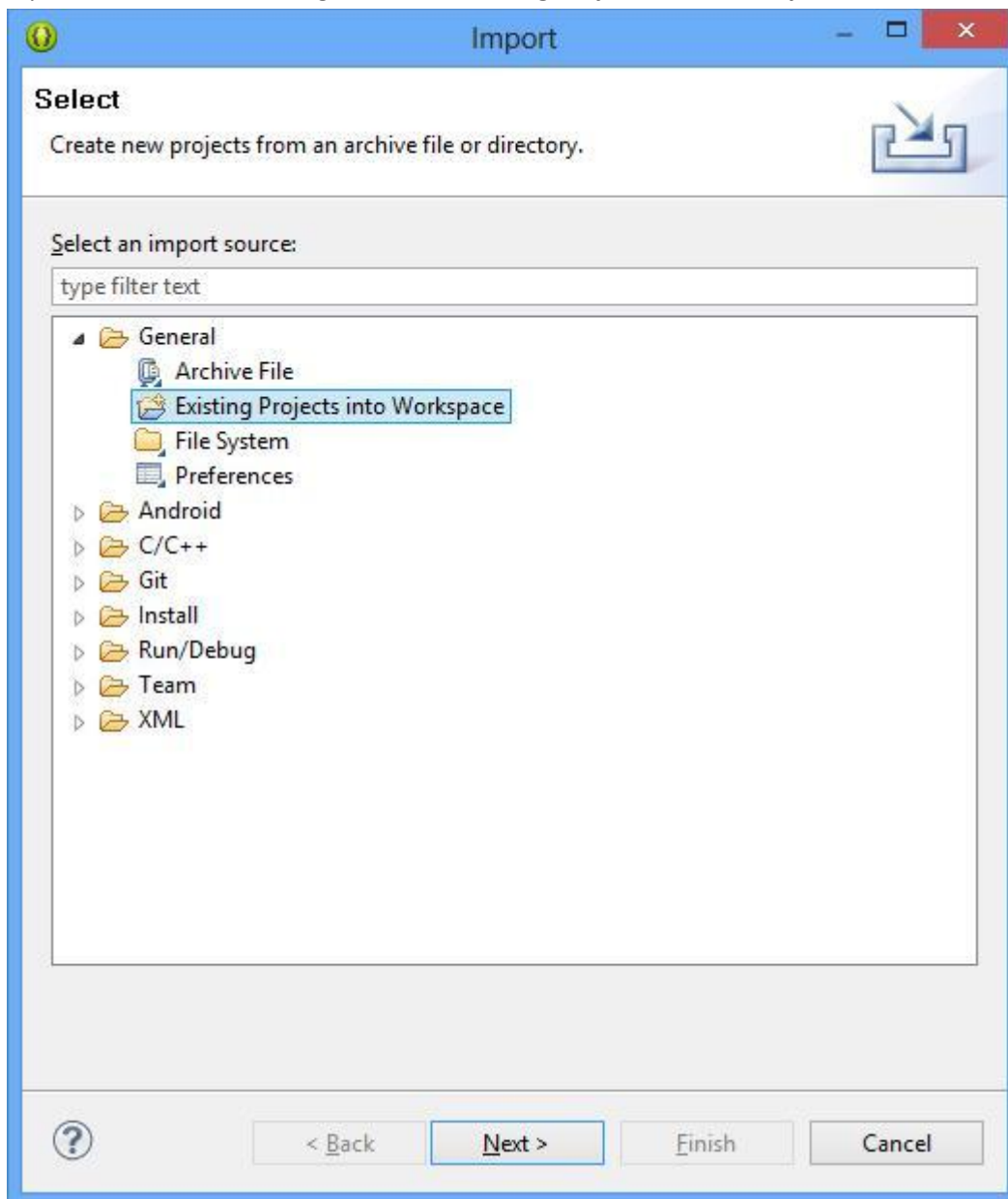
2. Expand the **General** heading and select **File System**
3. Click **Next** and select location where you would like to save exported project

Importing a Java project into Eclipse

1. To import an existing project into Eclipse, right-click on an empty space in Project Explorer and select **Import**.



2. Expand the **General** heading and select **Existing Projects Into Workspace**



3. Use the **Browse** button to find the project you would like to import

