Using Eclipse for Java

Using Eclipse IDE for Java Development

- Download the latest version of Eclipse from the website `http://www.eclipse.org`. It is available for various MS Windows OSes, Linux and MacOS. If you have don't have Java installed on your system, you would need to first install Java Development Toolkit (JDK) from `http://java.sun.com`.
- Eclipse is already installed in the `onyx` lab. There should be an eclipse icon on your desktop. Or just type `eclipse` on the console prompt.
- Eclipse comes with built-in tutorials and extensive help. Many good tutorials can also be found on the web.

Creating a new Java Project

- Click on `File` → `New` → `Java project`.

Creating a new Java Project

- Click on `File` → `New` → `Java project`.
- Choose the name of the project. The project contents can be left as default. In that case, Eclipse creates a folder under your default workspace folder with the same name as the project. For now choose the option "Use project folder as root for source and class files." Select *Next* to go to the next window. Then select `Finish` at the bottom to finish creating the project.

Creating a new Java Project

- Click on `File` → `New` → `Java project`.
- Choose the name of the project. The project contents can be left as default. In that case, Eclipse creates a folder under your default workspace folder with the same name as the project. For now choose the option "Use project folder as root for source and class files." Select *Next* to go to the next window. Then select `Finish` at the bottom to finish creating the project.
- To create a new class file, right click on the project pane (on the left) and choose the appropriate option. Click on the box for adding a `main` method if you need one in that class.

Handy Tips

- *Content Assist*. Use the keys `Ctrl-Space` to ask for help with function names, arguments and other topical content assistance.

Handy Tips

- *Content Assist*. Use the keys `Ctrl-Space` to ask for help with function names, arguments and other topical content assistance.
- *Word Completion*. Use the keys `Alt-/` to complete words after you type in the first few characters. Very useful to avoid having to type long variable or function names.

Handy Tips

- *Content Assist*. Use the keys `Ctrl-Space` to ask for help with function names, arguments and other topical content assistance.
- *Word Completion*. Use the keys `Alt-/` to complete words after you type in the first few characters. Very useful to avoid having to type long variable or function names.
- *Quick Fix menu*. Hover your mouse over an error and a quick fix menu drops down. Often the first suggestion will fix your error correctly! It will also help you find and insert the correct import statements when you classes from the Java library.

Handy Tips

- *Content Assist*. Use the keys `Ctrl-Space` to ask for help with function names, arguments and other topical content assistance.
- *Word Completion*. Use the keys `Alt-/` to complete words after you type in the first few characters. Very useful to avoid having to type long variable or function names.
- *Quick Fix menu*. Hover your mouse over an error and a quick fix menu drops down. Often the first suggestion will fix your error correctly! It will also help you find and insert the correct import statements when you classes from the Java library.
- *Automatic Javadoc comments*. Check options under the Source menu for options to automatically generate javadoc comments for classes and methods.

Handy Tips

- *Content Assist*. Use the keys `Ctrl-Space` to ask for help with function names, arguments and other topical content assistance.
- *Word Completion*. Use the keys `Alt-/` to complete words after you type in the first few characters. Very useful to avoid having to type long variable or function names.
- *Quick Fix menu*. Hover your mouse over an error and a quick fix menu drops down. Often the first suggestion will fix your error correctly! It will also help you find and insert the correct import statements when you classes from the Java library.
- *Automatic Javadoc comments*. Check options under the Source menu for options to automatically generate javadoc comments for classes and methods.
- *Formatting your program*. Check the *Format* option under the *Source* menu to automatically format your program nicely!

Building, Running and Debugging a New Project

- Every time you save your Java class file, Eclipse automatically compiles it for you.
- To run your Java program inside Eclipse, click on the *Run* menu and then choose Run (keyboard shortcut: `Ctrl-F11`) or Run Last Launched to run your Java program.
- You can also run your java program from the terminal directly by going to the folder that contains your project. You can also submit your assignment from the eclipse project folder.
- Click on *Run* $\rightarrow$ *Run Configurations*. Then a new window pops up that allows you create and manage run/debug configurations. For example, command line arguments can be set in the *Arguments* tab.

Importing Existing Java Classes into Eclipse

Suppose you have an exisiting Java program in a folder. Then you can bring it into Eclipse two different ways.

- **Import existing files into Eclipse project**. Create a new Java project in Eclipse. Click inside the project on the left pane. Then select *Import...* and then *General* and then *File System*. Then browse to the folder that contains your Java files and select the ones you want to import into your project. This will make a copy of those files into your eclipse workspace folder for the new project.

Importing Existing Java Classes into Eclipse

Suppose you have an exisiting Java program in a folder. Then you can bring it into Eclipse two different ways.

- **Import existing files into Eclipse project**. Create a new Java project in Eclipse. Click inside the project on the left pane. Then select *Import...* and then *General* and then *File System*. Then browse to the folder that contains your Java files and select the ones you want to import into your project. This will make a copy of those files into your eclipse workspace folder for the new project.
- **Create Eclipse project in existing folder**. Create a new java project in Eclipse. Then choose the option *Create project from existing source* and browse to the folder that contains your Java files. Then select *Finish* and now an Eclipse project has been created into your pre-existing folder!