CS 2100 - Coding Style Guide

In real-world software development, it is paramount to create readable and easily maintainable code. That is typically achieved through the use of style and commenting guidelines. Python is a language that forces good indentation practices, while Java does not. Therefore, we strongly encourage you to follow the following style and commenting conventions. We have also provided an Eclipse preferences file that sets up many of these formats automatically and will adjust your code using the **Format** command in the **Source** menu. (See the tutorial at the end of this document for more information.)

- **Filename:** The filename must match the containing class name and be **TitleCased**

- **Casing:** Class names should be **TitleCased** ; variable, field, and method names should be **camelCased**; constants should be **ALL_CAPS**

- **Indentation:** Code blocks should be indented **4** spaces using "soft tabs" (specifically, no hard **tab** characters). Our style file will automatically update Eclipse so that the **tab** key on your keyboard inserts 4 spaces automatically.

- **Curly braces:** Curly braces should open at the end of the line preceding the code block and end on a line of their own unless followed by an **else** or **catch** state- ment, such as:

```
if (true) {
// do something
}
if (test) {
// do something
} else {
// do something else
}
```

- **Header:** All files should contain the following heading, matching the homework number, your name, your userid, and listing any sources that may need to be cited. Please remember the academic integrity policy and the single source policy!

```
/**
 * Week N, Homework {1 or 2}
 * Your Name, Your Computing ID
 *
 * Sources: TA office hours, Big Java book, ...
 */}
```

- **Method and Field Comments:** Each field and method should be commented with information about that field or method, using the format below. You must also include the attribute information for parameters, **@param**, and return values, **@return** as shown. (This formatting is JavaDoc; these comments can be used to build documentation similar to that which Oracle provides for Java.)

```
/**
 * Short description of the field title
 */

private String title;

/**
 * Brief description of the method and what it should be
 * doing.
 *
 * @param x Describe what x is
 * @param y Describe what y is
 * @return What gets returned: what does this boolean mean?
 */
public boolean methodName(int x, float y) {...}
```

- **Inline Comments:** If there is a portion of your code that is difficult to understand, or you've received (collaboration policy appropriate) help on a portion of code, you should describe that in comments within your code. A good rule of thumb is that if you have fewer than one line of comment for every five to six lines of code, you have not commented enough.
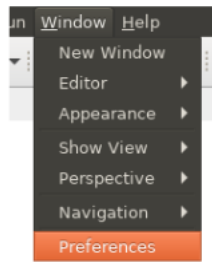
```
// Check that the new title is not null (edge case)
if (newTitle != null) { // Asked TA about edge cases
    this.title = newTitle;
}
```
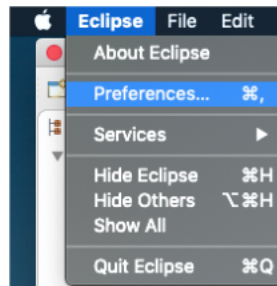
Installing CS 2100 Styles

To install the CS2100 Styles, download **CS2100Style.xml** from Collab under Resources. Then follow the steps below:

1. Open Eclipse preferences. On Windows or Linux, in the Window menu, choose Preferences. On Mac, in the Eclipse menu, choose Preferences.
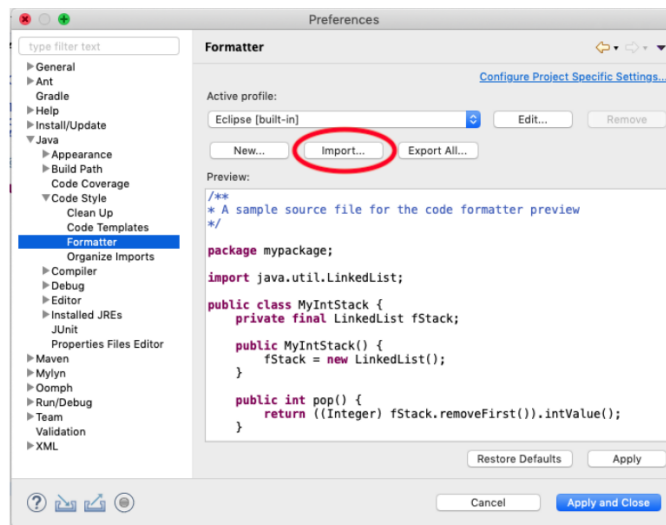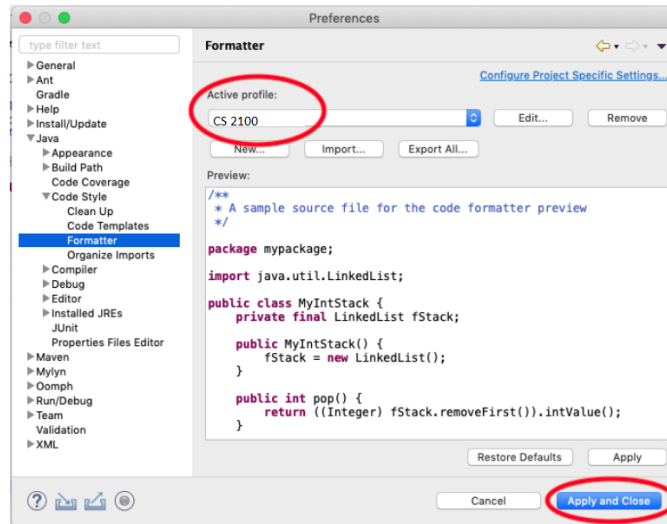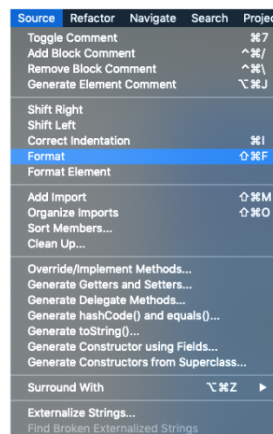

Windows/Linux   Mac

2. Use the dropdown menus on the left to choose Java, Code Style, Formatter, as shown below. (You may also search "Formatter" in the search box at the top left. By default, Eclipse uses the "Eclipse [built-in]" formatter. Choose "Import" and select the **CS2100Style.xml** file you saved from Collab.

3. Once chosen, you'll see that Eclipse has set the Active profile to "CS2100" as shown below. Click "Apply and Close" to begin using the new profile for code formatting.

4. To format your code, select all (**Ctrl+A** or **Command+A**) and choose Format from the Source menu, as shown below. You may also use the keyboard shortcuts **Ctrl+Shift+F** or **Command+Shift+F**

5. All of your code should be indented and formatted correctly. You will still need to be mindful of comments and naming conventions.