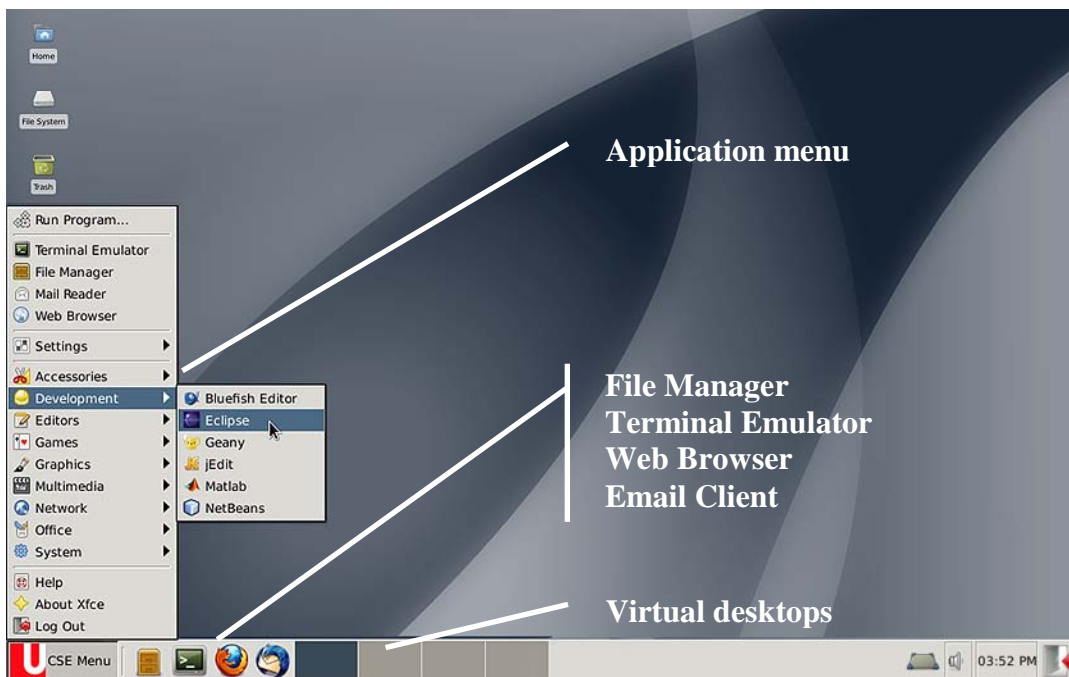# Guided Tour (Version 3.2)
By Steven Castellucci

This document was inspired by the Guided Tour written by Professor H. Roumani. His version of the tour can be accessed at the following URL: http://www.cse.yorku.ca/~roumani/jbaYork/GuidedTour.pdf.

The Desktop

Applications Menu

For a list of available applications, click the "CSE Menu". An application can be started by selecting it from the menu or entering a command in the terminal emulator. You can also log-out by selecting `Log Out` at the bottom of the menu.

Terminal Emulator

The terminal emulator (also known simply as "the terminal", "the console" or "the command-line") allows you to enter commands. You can use the terminal to make directories, move and copy files, and run applications. It is the most versatile operating system component that you will use in computer science. Please refer to the section Simple Terminal Commands.

Virtual Desktops

You can arrange you windows across four virtual desktops. Although you can only work with one at a time, the applications on all desktops remain running. Scrolling the mouse wheel on the desktop will display the virtual desktops in sequence. (You can also click on the thumbnail images along the bottom of the screen.) Even if you do not use the virtual desktops, make sure that you do not accidentally switch to them as you work.
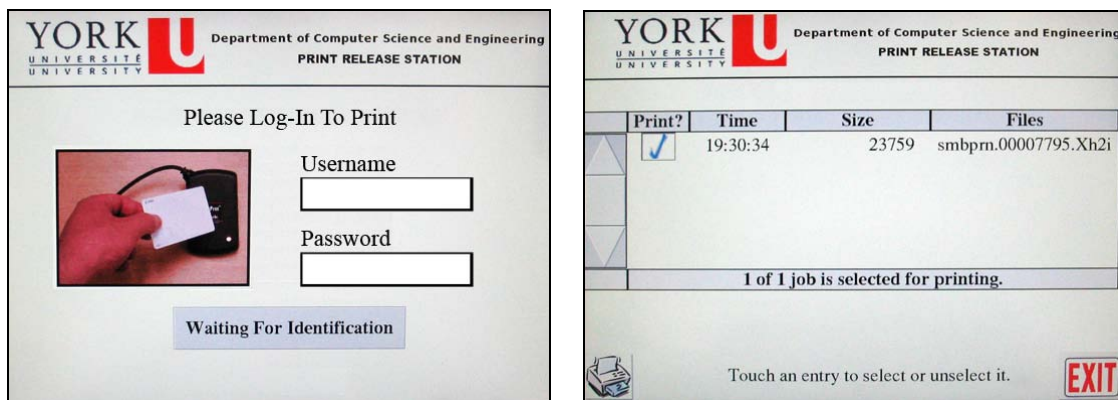
Mail Forwarding

You can have your Department mail automatically forwarded to the email account of your choice (e.g., yorku.ca, gmail.com). To do so, open a web browser and go to `mail.cse.yorku.ca`. Log-in using your EECS username and password. From the left pane, select `Options`, then `Mail`. From the center pane, select `Filters`, `Edit your filter rules`, then `Forward`. Enter your other email address(es) and click `Save`. To test the forwarding, send an email to your CSE address (e.g., *username*@cse.yorku.ca) and check your other email address(es).

Printing Files

You have a print quota of 500 pages. To display the number of pages remaining in your quota, enter the command `pquota` using the terminal.

Typically, you can print an open file by selecting `Print` from the `File` menu. After you send the file to be printed, go to any print station in room CSEB 1002, 1004, or 1006. Touch the screen to activate it. Enter your username and password using the attached keyboard. Ensure that the files you want printed are selected, and press the print icon in the bottom-left corner. To exit without printing, press the exit icon. If you experience any printing problems, contact the lab monitor.
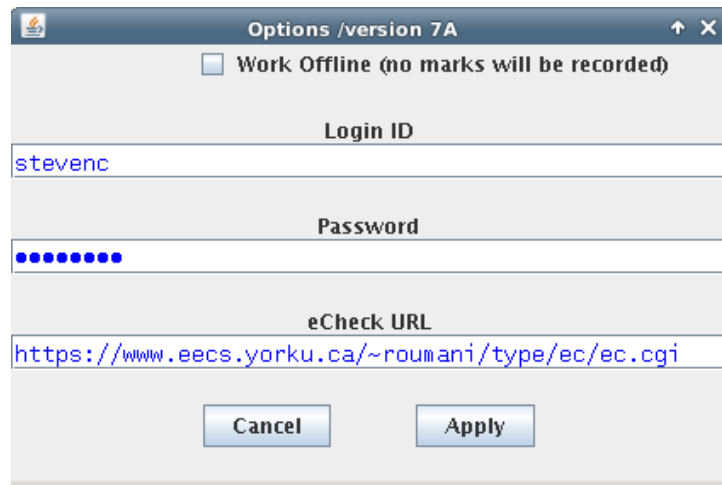
Configuring *eCheck*

Each chapter of the textbook contains programming exercises called "eChecks" (programs that are checked electronically). However, you must first configure the *eCheck* program that checks your code. Enter the command `java Options` at a terminal.

Uncheck "Work Offline". Enter your username, password, and the following URL:

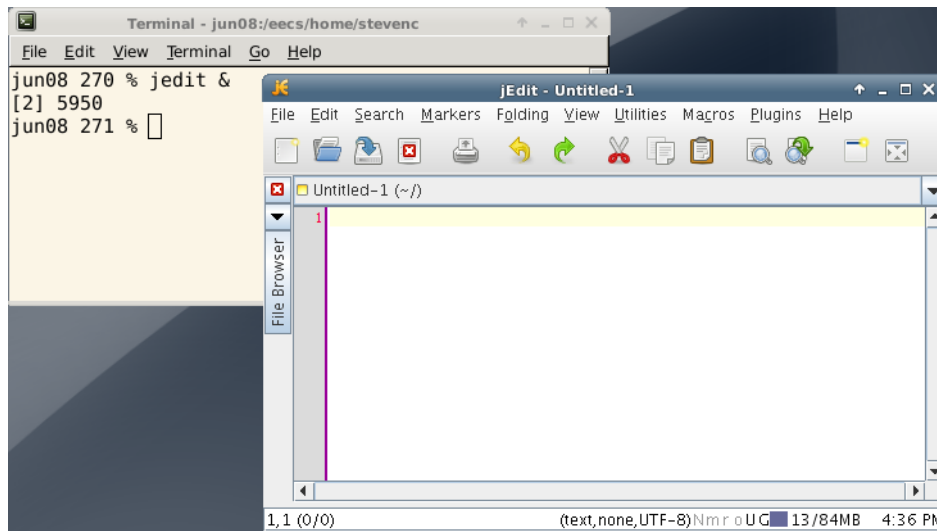`https://www.eecs.yorku.ca/~roumani/type/ec/ec.cgi`

Double-check that you entered the URL correctly. When you are finished, click `Apply`.
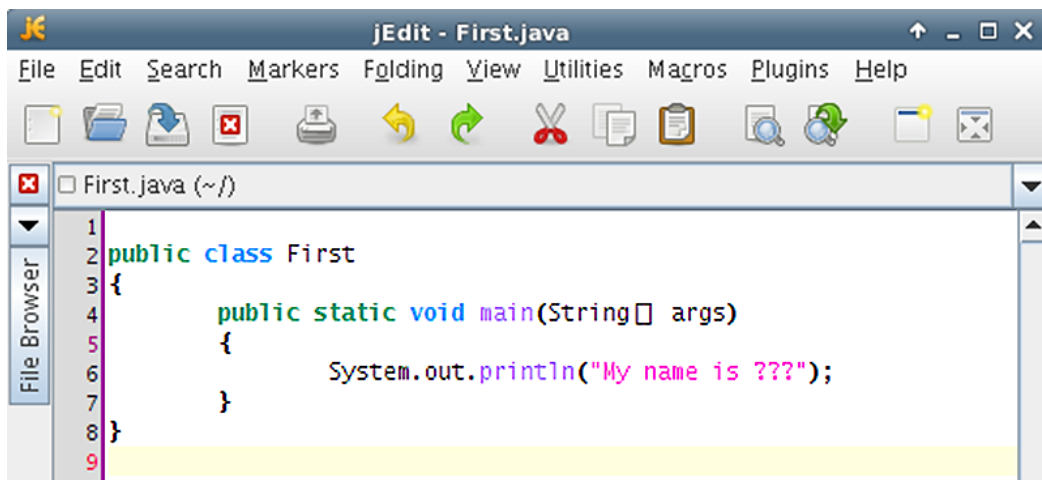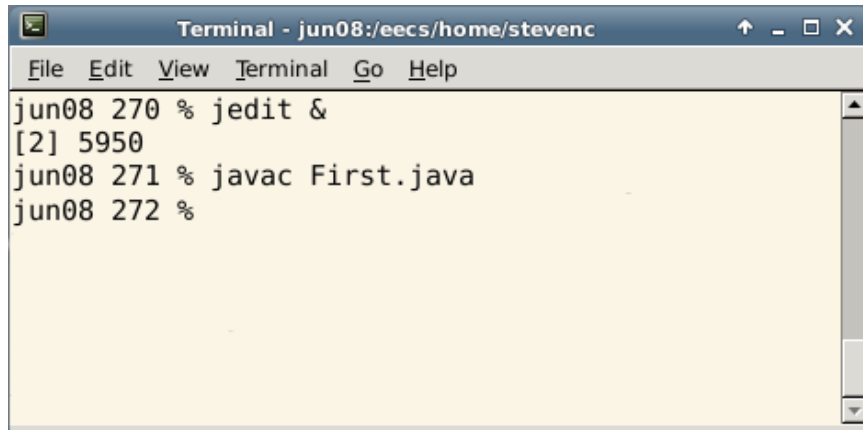
Creating a Java Program

Open a terminal emulator by clicking on its icon at the bottom of the screen. Type `jedit &` and press `Enter` on the keyboard. Including the ampersand (&) will allow to you to use the terminal while jEdit is running.

Type the following code in the jEdit window and save the file as `First.java`.
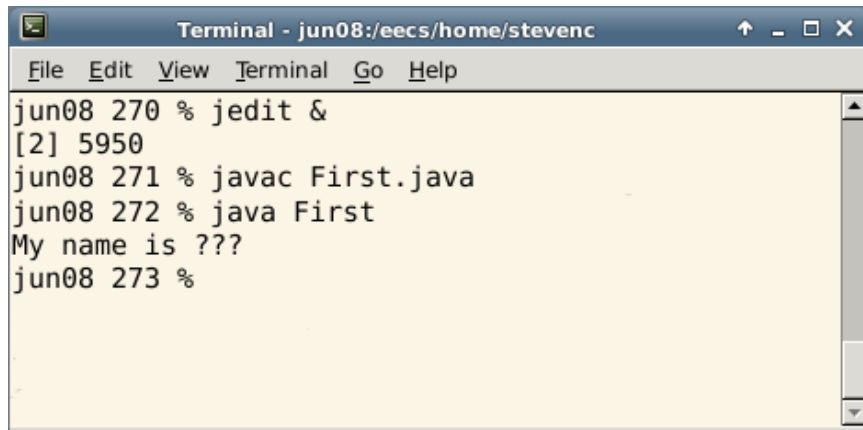
Back at the terminal, compile your program by entering the command `javac First.java`. If there are any errors outputted to the screen, return to jEdit and correct them. Make sure you save your changes and compile your code again.
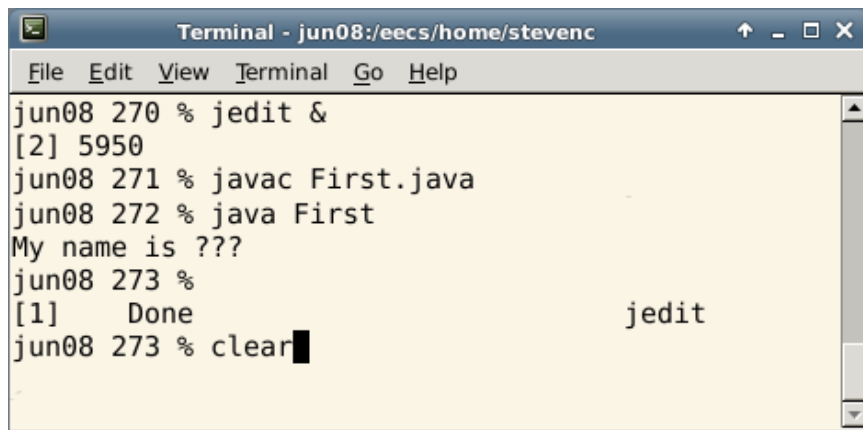
```
Terminal - jun08:/eecs/home/stevenc          ↑ _ □ X

 File  Edit  View  Terminal  Go  Help

jun08 270 % jedit &
[2] 5950
jun08 271 % javac First.java
jun08 272 %
```

Once your code compiles, run your program by entering the command `java First` at the terminal. Note that the command is `java`, **not** `javac`. Furthermore, note that the `.java` **file extension is omitted**. The output should be "My name is ???".
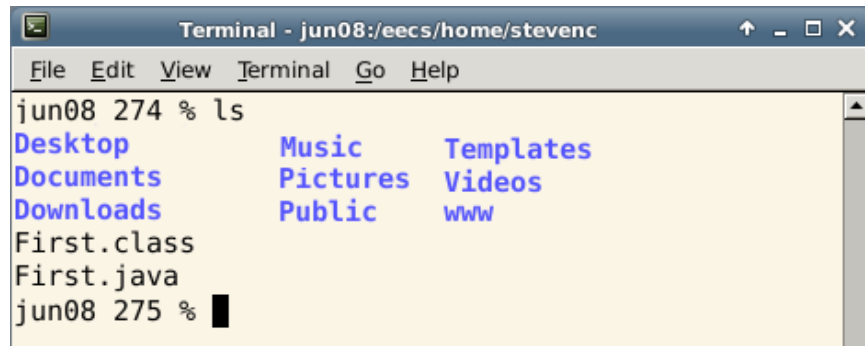
```
Terminal - jun08:/eecs/home/stevenc          ↑ _ □ X

 File  Edit  View  Terminal  Go  Help

jun08 270 % jedit &
[2] 5950
jun08 271 % javac First.java
jun08 272 % java First
My name is ???
jun08 273 %
```

Close the jEdit window. Before we change the program to output your username, let us create a directory in which to organize your work. Enter the command `clear` to clear the current contents of the terminal window.
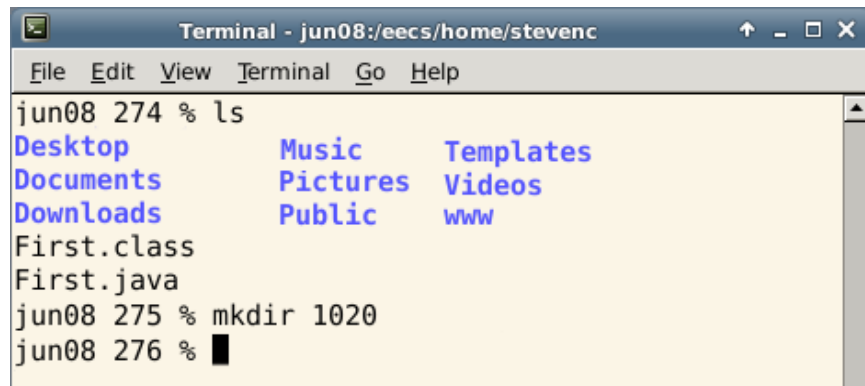
```
Terminal - jun08:/eecs/home/stevenc          ↑ _ □ X

 File  Edit  View  Terminal  Go  Help

jun08 270 % jedit &
[2] 5950
jun08 271 % javac First.java
jun08 272 % java First
My name is ???
jun08 273 %
[1]    Done                        jedit
jun08 273 % clear█
```

Enter the command `ls` to **lis**t the contents of the current directory. Directories appear in purple, files typically appear in black. In this directory are (at least) two files: the source file and the compiled class file.
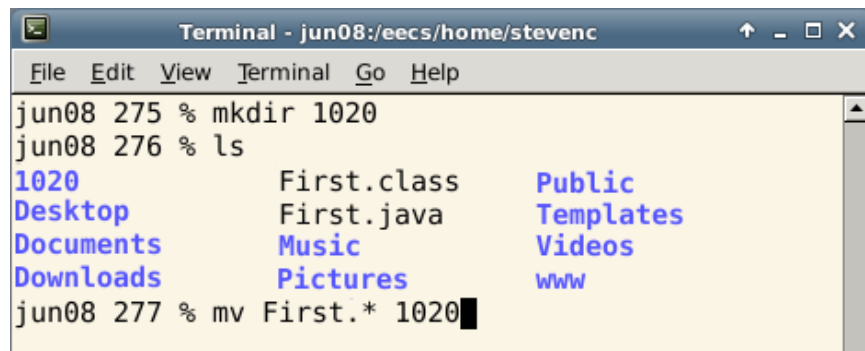
```
Terminal - jun08:/eecs/home/stevenc           ↑ _ □ ✕
 File  Edit  View  Terminal  Go  Help
jun08 274 % ls
Desktop        Music      Templates
Documents      Pictures   Videos
Downloads      Public     www
First.class
First.java
jun08 275 % █
```

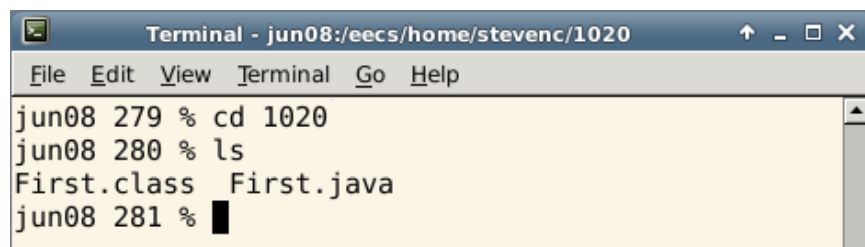Make a directory called "1020". To do this, enter `mkdir 1020` at the terminal.

```
Terminal - jun08:/eecs/home/stevenc           ↑ _ □ ✕
 File  Edit  View  Terminal  Go  Help
jun08 274 % ls
Desktop        Music      Templates
Documents      Pictures   Videos
Downloads      Public     www
First.class
First.java
jun08 275 % mkdir 1020
jun08 276 % █
```

Enter the command `ls` again to see that the directory was created. Now, move both files to the 1020 directory. Because both files have similar names, we can use the asterisk wildcard (*) to represent both file extensions. Enter the command `mv First.* 1020`. The first argument represents the files to move, and the second argument represents the destination directory.
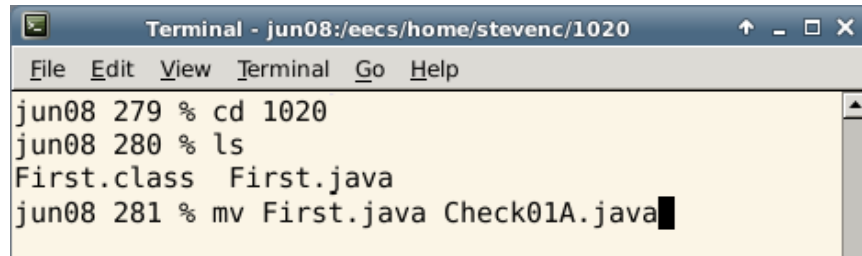
```
Terminal - jun08:/eecs/home/stevenc           ↑ _ □ ✕
 File  Edit  View  Terminal  Go  Help
jun08 275 % mkdir 1020
jun08 276 % ls
1020           First.class    Public
Desktop        First.java     Templates
Documents      Music          Videos
Downloads      Pictures       www
jun08 277 % mv First.* 1020█
```

Commands entered at the terminal typically affect the current directory, which appears in the title bar. Enter the command `cd 1020` to change the current directory (your home directory) to the 1020 directory. Notice the change in the title bar. Enter `ls` to see the contents of this directory. The two "First" files should be there.
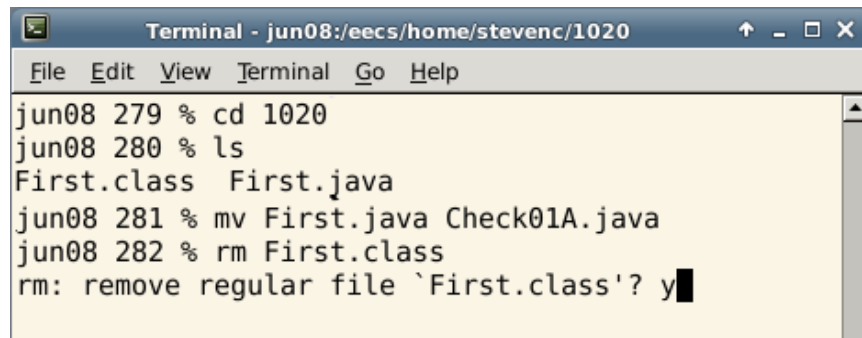
```
Terminal - jun08:/eecs/home/stevenc/1020       ↑ _ □ ✕
 File  Edit  View  Terminal  Go  Help
jun08 279 % cd 1020
jun08 280 % ls
First.class  First.java
jun08 281 % █
```

Rename the `First.java` file to be `Check01A.java`. In Linux, the `mv` command is used to move files, as well as rename them. Enter the command `mv First.java Check01A.java`. In this case, the first argument represents the current filename, and the second argument represents the new filename.
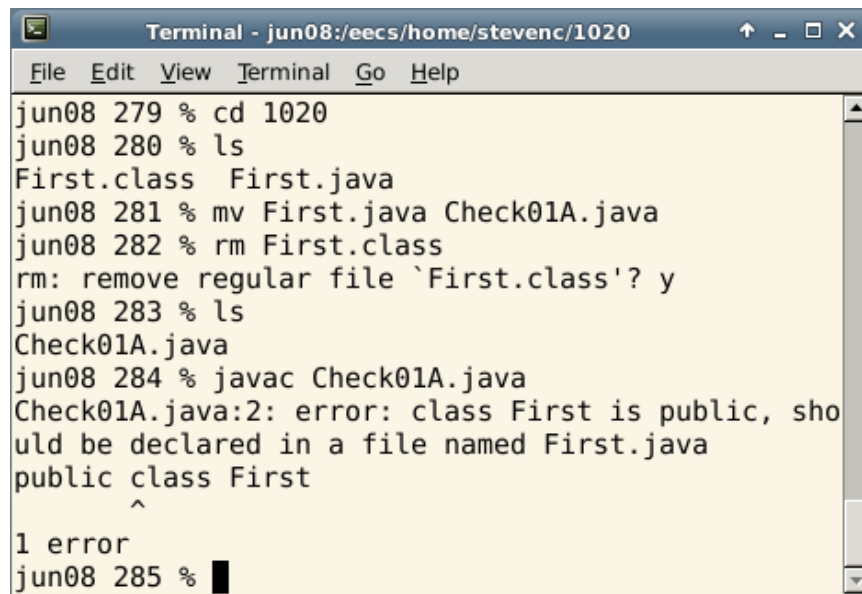
```
Terminal - jun08:/eecs/home/stevenc/1020

File  Edit  View  Terminal  Go  Help

jun08 279 % cd 1020
jun08 280 % ls
First.class  First.java
jun08 281 % mv First.java Check01A.java
```

Now that the file is renamed, the old class file is no longer needed. Delete it using the command `rm First.class`. **Use the** `rm` **command with caution**, as files **are not** moved to a Trash or Recycle Bin; **they are deleted permanently**.

```
Terminal - jun08:/eecs/home/stevenc/1020

File  Edit  View  Terminal  Go  Help

jun08 279 % cd 1020
jun08 280 % ls
First.class  First.java
jun08 281 % mv First.java Check01A.java
jun08 282 % rm First.class
rm: remove regular file `First.class'? y
```
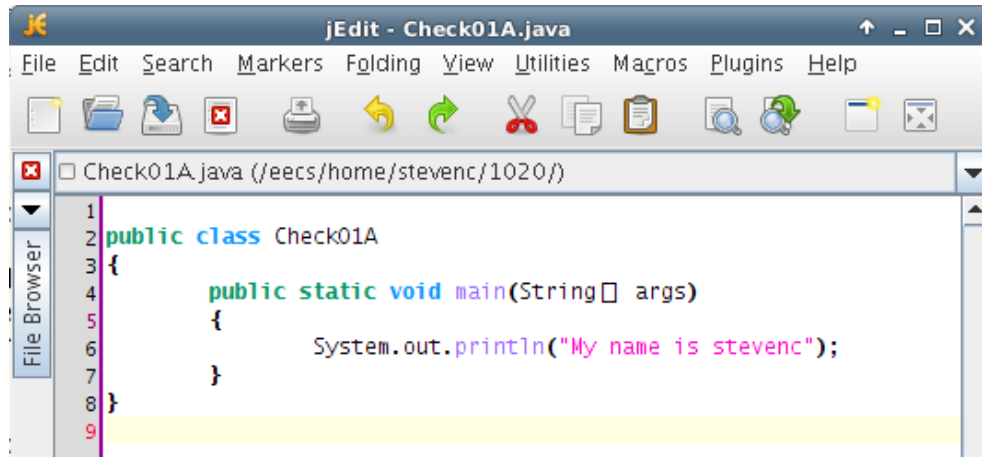
Try compiling the `Check01A` class. You will receive a syntax error, stating that the filename should be changed to the class name.

```
Terminal - jun08:/eecs/home/stevenc/1020

File  Edit  View  Terminal  Go  Help

jun08 279 % cd 1020
jun08 280 % ls
First.class  First.java
jun08 281 % mv First.java Check01A.java
jun08 282 % rm First.class
rm: remove regular file `First.class'? y
jun08 283 % ls
Check01A.java
jun08 284 % javac Check01A.java
Check01A.java:2: error: class First is public, sho
uld be declared in a file named First.java
public class First
       ^
1 error
jun08 285 %
```
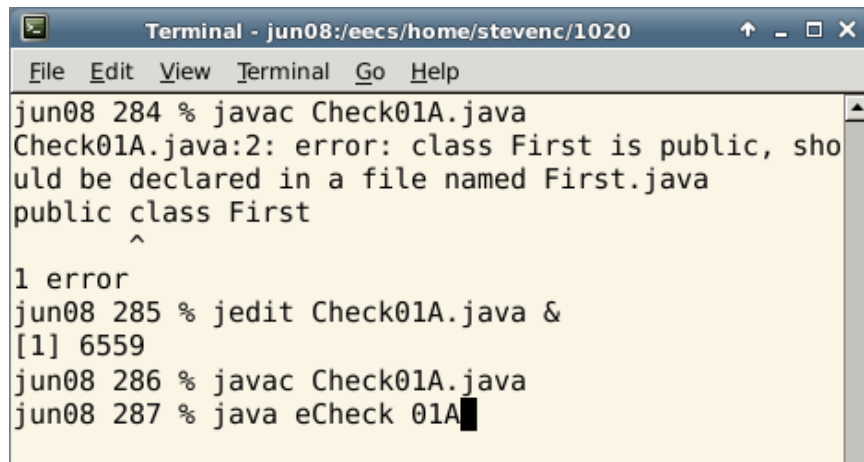
Open jEdit again to edit the source code. This time, use the command `jedit Check01A.java &` to start jEdit and automatically open the file `Check01A.java`. Again, the use of the ampersand (&) allows the terminal window to be used while jEdit is running.

Make two changes to the source code: (1) change the name of the class to Check01A; and (2) replace "???" with your EECS username.
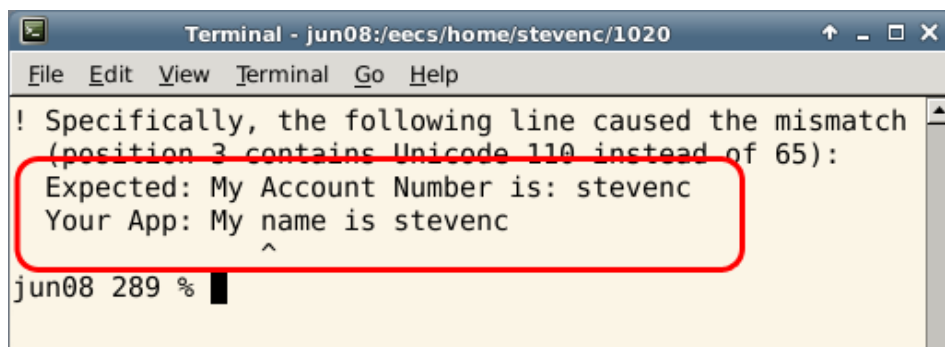

Compile your code again. Correct any syntax error that exist, save your changes, and recompile. When there are no compilation errors, check your program using eCheck. The command to run eCheck is `java eCheck ###`, where "###" is replaced with the code for that eCheck exercise. For this exercise, the code is "01A".
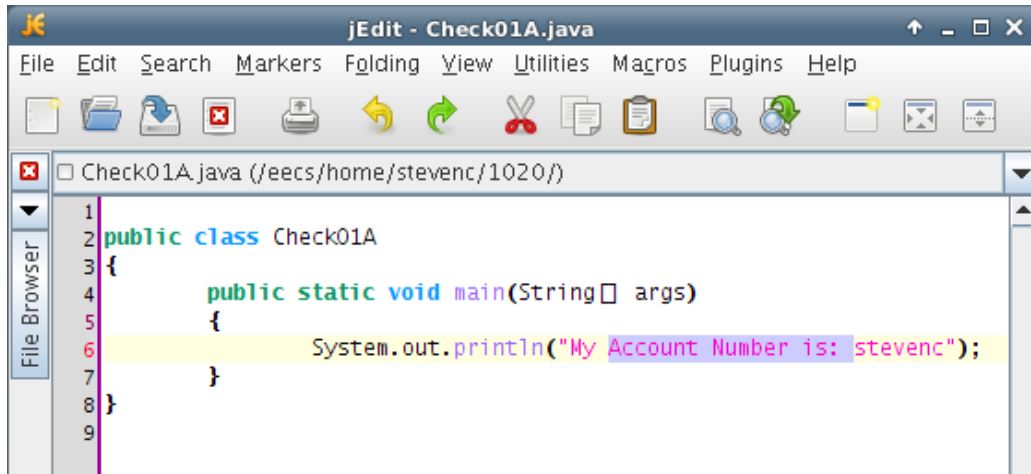

The eCheck output will indicate that there is a discrepancy in your program's output. In this case, "name" should be replaced with "Account Number".
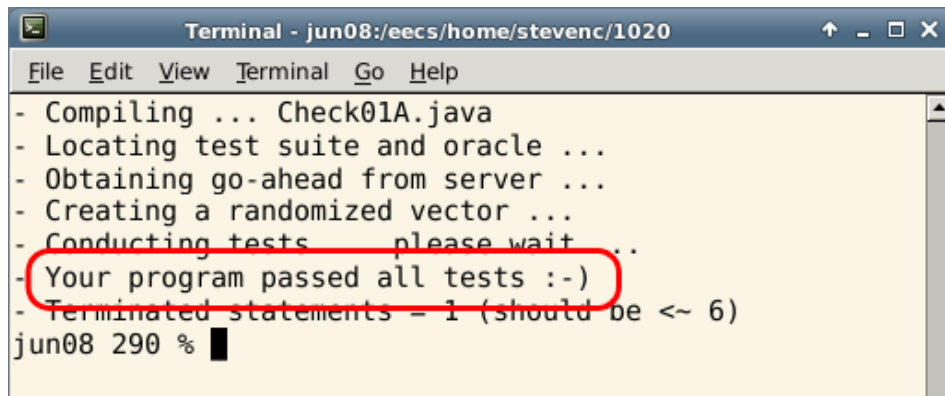
Return to jEdit and make the required changes to your code. Save the changes, recompile your code, and run eCheck again.

The eCheck program will indicate when your program has passed all tests. If your program's output does not match the expected output, identify the discrepancy, change your code to correct the mismatch, and eCheck your program again.

Auto-Completion and Command History

You do not have to type entire filenames or directory paths. Type the first couple of characters, followed by the `Tab` key. The operating system will complete the rest of the name or path. If there are multiple matches, the operating system will complete only the common portion. You will have to type additional characters to identify the desired file or directory. To repeat a command at a terminal, you can use the up- and down-arrow keys to cycle through commands you previously entered.

Simple Terminal Commands

Command: `man command`   **Example**: `man submit`
Description: Displays the user manual for the passed command. The user manual details the type and number of arguments required by the command, and lists all the available command options. To scroll through the user manual, press the spacebar. To exit the user manual, simply press the Q-key.

Command: `mkdir dirName` **Example**: `mkdir eChecks`
Description: Creates a subdirectory with the passed name in the current directory. The example creates a subdirectory called "eChecks".

Command: `cd dirName`   **Example 1**: `cd`   **Example 2**: `cd ..`   **Example 3**: `cd mail`
Description: Without any argument (Example 1), this command changes the working directory to your home directory (equivalent to the "My Documents" folder in *Windows*). With the argument "`..`" (Example 2), this command changes the working directory to the parent directory. If you provide the name of a subdirectory as an argument (Example 3), this command changes the working directory to be that subdirectory (e.g., the subdirectory called "mail").

Command: `ls dirName`   **Example 1**: `ls`   **Example 2**: `ls mail` **Example 3**: `ls *.txt`
Description: Lists the contents of the directory specified by the argument. Without any arguments (Example 1), this command lists the visible contents of the working directory. If the argument is a directory name (Example 2), this command lists the visible contents of that directory (e.g., the subdirectory called "mail"). Example 3 lists all files in the current directory that have a ".txt" extension. You can use the "*" wildcard to search for files that match a pattern. There are many options for this command, such as "`-a`" to show hidden files and "`-l`" to show file and directory details. Enter the command `man ls` for further details.

Command: `rm fileOrDir`   **Example 1**: `rm First.java`   **Example 2**: `rm -r eChecks`
Description: Removes the file or directory indicated by the argument. The first example deletes the file "First.java". The second example (note the "-r" option) removes directory called "eChecks" and all of its contents. **Use this command with caution!**

Command: `cp orgnl cpy`   **Example**: `cp First.java First_backup.java`
Description: Copies the file *orgnl* to the location *cpy*. The example creates a copy of "First.java", called "First_backup.java".

Command: `mv old new`   **Example**: `mv First.java Second.java`
Description: Moves the file *old* to the location *new*. This command can also be used to rename files. The example renames "First.java" to "Second.java".