Introduction to Computer Science II
Lab 1
Simple chat bot

1 Getting started

These instructions (and all of the assignments that follow) assume that you've installed the necessary tools. If you haven't done that, go back to the course web page, find the *Tools* document, and follow its instructions. When you're done, return here.

With the tools installed, grab a source code file with which to work…

1. Open your *Terminal* app. On Mac or Windows, just search for `Terminal` and you will find it. A new window should open and leave you at a *shell prompt*. This prompt typically ends with a dollar sign (`$`), and we will henceforth use that symbol to represent the prompt itself. (Which is to say, don't type the prompt when you see it in the instructions; it's just an indicator that you're expected to type a command at the shell prompt.)

2. Create a folder (a.k.a., a *directory*) for this lab, and then change into that directory:

   ```
   $ mkdir lab-1
   $ cd lab-1
   ```

   You may see the shell prompt change to show that you are now in that `lab-1` directory.

3. **Get the code:** At the prompt, enter the following command to grab the source code file for this lab:

   ```
   $ curl -LO https://sfkaplan.people.amherst.edu/courses/2024/fall/
              COSC-112/assignments/lab-1/SimpleChat.java
   ```

4. **Examine the code:** Open `SimpleChat.java`[1]—be careful to note the *space* and the *dot* that follow `code` in this command:

   ```
   $ code .
   ```

5. **Compile and run:** Use these command first to compile and then to run the code you downloaded and examined:

[1] These instructions assume that you are using *vscode*. If you are using something else, feel free, but you're on your own.

```
$ javac SimpleChat.java
$ java SimpleChat
```

If you have questions about file management with the terminal, I would recommend starting by checking out this video tutorial:

`https://www.youtube.com/watch?v=i16cQTULEjM`.

2 Your assignment

In this lab you will be editing the main method of a SimpleChat class to create an interactive chat bot. This method will use a `Scanner`. For documentation on how `Scanner` works, and to see what other methods are available, see the Java API documentation:

`https://docs.oracle.com/en/java/javase/21/docs/api/index.html`

On that page, in the upper right, you will find a *Search* textbox. In it, enter `Scanner`, and a drop-down menu will show you all kinds of things that contain that string. The one we want is at the top, `java.util.Scanner`, so select it. Doing so will bring up a new page that shows a description of this class, as well as all of the methods that go with it. Using this kind of documentation about classes and objects will become a valuable part of working with Java.

Completing the lab: Here's what you must do...

1. **Create at least five conversational prompts.** These prompts can be whatever you'd like as long as they appropriately respond to user input.

2. **Set an exit keyword.** If the user types the input of "exit" at any point, no other prompts should run and the program should terminate.

3. **Enable shape printing in your chatbot.** When the user asks to see a *square*, *triangle*, or *tree* of a certain size, the chatbot should print out the corresponding pattern. The *size* indicates the number of + signs on each edge of a square, the number of + signs on each of a right triangle, and the number of + bbbbbbbbb*she signs constituting the height of three. For example, here is a square of size 6:

```
++++++
+    +
+    +
+    +
+    +
++++++
```

Here is a triangle of size 6:

```
+
++
+ +
+  +
+   +
++++++
```

And, here is a tree of size 6:

```
     +
    +++
   +++++
  +++++++
 +++++++++
+++++++++++
     |
```

These images should be printed out whenever the corresponding keyword and size are included in the user's prompt, e.g. "show a tree of size 3 please" and "Display tree size 3." should result in the same image. You can assume the use of the size keyword preceding the size integer.

Note: You are able to create as many prompts manually as you'd like, but can also create iterative prompts using loops. The use of loops (and nested for loops in particular) will be especially important in the creation of the shapes. You may want to consider the use of a while loop to assess for and implement the exit keyword.

3 How to submit your work

In order to submit your assignments, we will use a shared folder in *Google Drive*. If you have registered for the class, we will have created and shared with you a folder that is shared with the faculty and main TAs for the course. Within that folder, we will create a subfolder for each assignment. Submitting your work for each assignment will simply require you to copy your source code file(s) into that assignment's subfolder.

For this assignment, you need to copy your completed `SimpleChat.java` file into the `lab-1` subfolder within your shared Google Drive folder.[2] To do so, follow these steps:

[2] Be sure to submit the `.java` source code file, and **not** the `.class` file that is produced when your code is compiled by the `javac` command.

1. **Open your shared folder:** You should have received an email with a shared Google Drive folder, and that will provide a link to that folder, opening it in your browser; alternatively, you can go to drive.google.com, and on the left side of the window, select *Shared with me* to see that folder (amongst any other files and folders shared with you).

2. **Open the subfolder:** There should be a folder named `lab-1` in the shared folder; open it within the browser.

3. **Open your local folder:** On your own computer, open your local `lab-1` folder/directory where you've been doing your programming in the *Finder/Explorer* (for *macOS* and *Windows*, respectively). You can do so from the *Terminal*, where the shell prompt is already in your `lab-1` directory, like this for *macOS*...

   ```
   $ open .
   ```

   ...and like this for *Windows*...

   ```
   $ explorer.exe .
   ```

4. **Copy:** Drag your `SimpleChat.java` source code file from your *Finder/Explorer* window into your Google Drive window in your browser. That will upload the file into that shared folder for `lab-1`.

This assignment is due on Sunday, Sep-15, 11:59 pm.