

# INTRODUCTION TO COMPUTER SCIENCE I

## LAB 3: PROVIDING WISDOM

Friday, February 9, 2018

### 1 Providing user choices, but fancy-like

We have seen how to read in a value from the keyboard, and we will use that capability here. We may, however, want to be more clever about the options we present the user, and our interpretation of those values.

Imagine running a program named `Wisdom`, and seeing the following menu appear:

```
MENU:
    (  1) A joke
    ( 10) A haiku
   (100) A quote
  (1000) A koan
Enter a sum of the messages you want to see:
```

Now, you could enter any one of the choices and see the result:

```
Enter a sum of the messages you want to see: 10
-----
Zamboni summer
melts away last year's defeats --
fresh October ice
-----
```

But you may also **add** choices to select more than one:

```
Enter a sum of the messages you want to see: 101
-----
Alice: I know a great knock-knock joke.
Bob:   OK!
Alice: So, you start.
Bob:   Oh, so...Knock-knock?
Alice: Who's there?
Bob:   ...?!
-----
-----
You can observe a lot by just watching. --Yogi Berra
-----
```

How do we program such an interface?

## 2 Your assignment

Write a program, named `Wisdom`, as described in Section 1. That is, write a program that presents exactly the menu shown. The user should be able to enter a single integer that selects any subset of a joke, haiku, quote, and koan (but only one of each type; the user should only enter 4-digit numbers made up of 0's and 1's). Find or write your own joke, haiku, quote, and koan!

You will be creating the program for this assignment from scratch, rather than filling in part of a program that I've started for you. Section 3 reminds you of some useful terminal commands that you can use to make and navigate between directories, create new files, and compile and run your code.

There are many different ways to structure your code to complete this task. Some options involve a large number of if statements and checking different user inputs separately, others involve doing a bit of computation to allow you to write fewer if statements. Your goals are (1) to write a program that works correctly, and then (2) to come up with a solution that uses as few if statements as possible. My solution uses four if statements. Can you beat me?

Before you submit your program, be sure to *test it thoroughly*. When I grade this assignment, I will be looking for your program to work properly for any value the user could input (for now we will assume that our users are reasonable and will only enter a positive, 4-digit integer consisting of 0's and 1's). Think carefully about how you should test your program to convince yourself that it'll work for all cases!

## 3 Some useful terminal commands

To create a directory:

```
$ mkdir lab3
```

To change into a directory:

```
$ cd lab3
```

(For example, if your `lab3` directory is on your Desktop, then if you are in the Desktop, entering the command `cd lab3` will move you into your `lab3` directory.)

To move up a directory:

```
$ cd ../
```

(For example, if your `lab3` directory is inside your home directory, then if you are in the `lab3` directory, entering the command `cd ../` will move you up into your home directory.)

To list the contents of a directory:

```
$ ls
```

(Note that this is a lowercase letter L, not the number 1.)

To open/create a file:

```
$ emacs Wisdom.java &
```

(If the file `Wisdom.java` already exists, this command will open it in emacs. If the file does not

exist, this command will create a new file with the name `Wisdom.java`, and then open that new (blank) file in `emacs`. The `&` at the end of the line ensures that you can type commands into the terminal while your `emacs` window is still open.)

To compile your code:

```
$ javac Wisdom.java
```

(Remember to **SAVE** your work before issuing this command.)

To run your code:

```
$ java Wisdom
```

(Remember to **COMPILE** your code before issuing this command.)

## 4 Submitting your work

Submit your `Wisdom.java` file with the CS submission system, using one of the two methods:

- **Web-based:** Visit the submission system web page.
- **Command-line based:** Use the `cssubmit` command at your shell prompt:

```
$ cssubmit Wisdom.java
```

**This assignment is due on Thursday, February 15, 11:59 pm.**