Lab 2: Basic Flow Control

CSE/IT 107

NMT Computer Science

> “When you come to a fork in the road, take it.”
>
> — Attributed to Yogi Berra

> “Simplicity is the ultimate sophistication.”
>
> — Leonardo Da Vinci

> “How do we convince people that in programming simplicity and clarity – in short: what mathematicians call “elegance” – are not a dispensable luxury, but a crucial matter that decides between success and failure?”
>
> — Edsger Dijkstra

1 Introduction

The purpose of this lab is to introduce you to the fundamentals of flow control - if, else, and while loops.

2 Boolean logic, `if`, and `else`

A common activity when programming is determining if something value is true or false. For example, if a variable is less than five or if the user entered the correct password. Any statement that can be resolved into a true or a false value is called a boolean statement, the value it resolves into (true or false) is called a boolean value.

```
>>> x = 5
>>> print(x < 3)
False
>>> print(x < 6)
True
```

In the above example, the boolean values are `True` and `False`. The boolean statements are `x < 3` and `x < 6`.

In addition to `<`, we can also test for other inequalities.

```
>>> x = 3
>>> y = 6
>>> print(x < y)
True
>>> print(x > y)
False
>>> print(x <= y)
True
>>> print(x >= y)
False
```

Note that `<=` means "less than or equal to" and `>=` means "greater than or equal to".

Finally, we can test if two values are equal (`==`) or not equal (`!=`).

```
>>> x = 3
>>> y = 3
>>> z = 4
>>> print(x == y)
True
>>> print(x == z)
False
>>> print(y != 5)
True
>>> print(y != x)
False
```

It is important to remember that we use `=` to assign a value to a variable and `==` to test if two values are equal.

3 `while` loops

4 Turtle

5 Exercises

6 Submitting

You may submit your code as either a tarball (instructions below) or as a .zip file. Either one should contain all files used in the exercises for this lab. The submitted file should be named either `cse107_firstname_lastname_lab2.zip` or `cse107_firstname_lastname_lab2.tar.gz` depending on which method you used.

For Windows, use a tool you like to create a `.zip` file. The TCC computers should have `7z` installed.

Upload your tarball or .zip file to Canvas.

6.1 Linux

Tar is used much the same way that Zip is used in Windows: it combines many files and/or directories into a single file. Gzip is used in Linux to compress a single file, so the combination of Tar and Gzip do what Zip does. However, Tar deals with Gzip for you, so you will only need to learn and understand one command for zipping and extracting.

In the terminal (ensure you are in your `lab1` directory), type the following command, replacing `firstname` and `lastname` with your first and last names:

```
tar czvf cse107_firstname_lastname_lab2.tar.gz *.py
```

This creates the file `cse107_firstname_lastname_lab1.tar.gz` in the directory. The resulting archive, which includes every python file in your `lab1` directory, is called a tarball.

To check the contents of your tarball, run the following command:

```
tar tf cse107_firstname_lastname_lab2.tar.gz *.py
```

You should see a list of your Python source code files.