

# CIS 210, Fall 2016

## Introduction to Computer Science

### Main Menu

[Class home page](#)[Piazza](#)[How to Succeed](#)[Schedule](#)[Assignments](#)[References](#)[Exams](#)

### Project 2, Part 2

This assignment is due Friday, October 7 at 5pm. Save your Python program in a file called squareroot.py and turn that file in using Canvas.

### Purpose

Approximate the square root of a number using iteration.

### Approximation of the square root of a number

We have used the square root function from the math library a number of times in Chapter 2 of the textbook. This project is for you to write your own approximation function using the following equation:

$$X_{k+1} = \frac{1}{2} \times \left( X_k + \frac{n}{X_k} \right), \text{ where } X_0 = 1$$

Each value of X should be a better approximation for sqrt(n).

### Requirements

You will create a Python program to compute the square root of a number using the above equation.

```
$ python3 squareroot.py 123.45 -i 50
```

After 50 iterations, sqrt(123.45) = x.xxxxx

```
$ python3 squareroot.py 123.45 -i 99
```

After 99 iterations, sqrt(123.45) = x.xxxxx

After you have this working, you should compute the difference between the approximate value you have computed and that returned by the sqrt function in the math library, and indicate the percent error, assuming that the value returned by the math library function is correct. This version of the program should yield the following:

```
$ python3 squareroot.py 123.45 -i 50
```

After 50 iterations, sqrt(123.45) = x.xxxxx; this represents y.yy% error compared to the math library

### Getting started

This [starter code](#) may be helpful. You will also need the test\_harness.py library module from the [here](#).

### Grading

35 points possible

- 15: Program runs and consistently produces correct output for arbitrary positive numbers. 5 points if it is on the right track, but does not yield the correct responses.
- 10: Follows [CIS 210 coding guidelines](#), including author identification and header.
- 10: Clarity. The program should not only be consistent with the requirements and approach described here, but it should be very easy to read the program and verify its consistency with the spec.

