WIT COMP1000 Computer Science I

Prof. Thai

Student:

**Lab10: Square Root**

1. **SquareRoot** (SquareRoot.java)

Write a program to compute the square root of a number. DO NOT USE any math libraries/methods in this program. You will be using the Babylonian method (a.k.a. Heron’s method) to approximate the square root[[1]](#footnote-1).

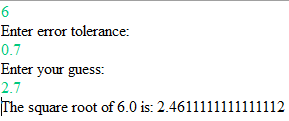
To calculate the square root of x, the Babylonian method requires three inputs: x, an initial guess for the square root, and the error tolerance. It uses a repetitive calculation to get closer and closer to the actual value of the square root: after each iteration, the method checks if the absolute value of the difference between **nextGuess** and **lastGuess** is less than the error tolerance; if so, it stops and returns the value of **nextGuess** as the square root, otherwise if the difference between **nextGuess** and **lastGuess** is larger than the error tolerance, it repeats the calculation. Here is an example (x is 16, the initial guess is 6, and the error tolerance is 0.5):

* The first iteration computes **nextGuess** to be (6+16/6)/2 = 4.3333.
* Next, check the difference between the **lastGuess** (6) and **nextGuess** (4.3333). The absolute value of the difference is 1.6667, which is greater than the error tolerance of 0.5. So, repeat the calculation.
* The second iteration computes **nextGuess** to be (4.3333+16/4.3333)/2 = 4.0128.
* Check the difference between **lastGuess** (4.3333) and **nextGuess** (4.0128). The absolute value of the difference is 0.3205, which is less than the error tolerance of 0.5. So, the procedure stops and returns 4.0128 as the square root of 16.

Your program needs three methods: **main()**, **squareRoot()**, and **absoluteValue()**. The **main()** method should get all three inputs from the user (x, initial guess, and error), run the **squareRoot()** method, and output the approximate value of the square root (using exactly five decimal places, rounding if necessary). As always, you must validate the user’s inputs – all three must be positive. If the user enters a value that is not positive, then your **main()** method should prompt them to enter the value again (and repeat until they enter a valid input).

As you cannot use any math libraries/methods, you will also need to write your own **absoluteValue()** method. Only the **main** method should interact with the user (getting inputs and displaying results). The **squareRoot()** and **absoluteValue()** methods must not contain any input/output statements.

**TAKE A SCREENSHOT** of the output console and paste it here. Do not paste your source code in this document.

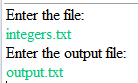


1. **StorePositiveIntegers** (StorePositiveIntegers.java)

Write a program that reads a stream of integers from a file and writes only the positive numbers to a second file. The user should be prompted to enter the names of both the input file and output file in **main()**, and then **main()** should attempt to open both files (providing an error if there is an error during this process). The **main()** method should then call the **processData()** method to read all the integers from the input file and write only the positive integers to the output file. The **processData()** method takes arguments: a **Scanner** object to read from the input file and a **PrintWriter** object to write to the output file. You can assume that if you are able to successfully open the input file, then there will only be integers in it.

Create an input file called *integers.txt* that contains both positive and negative whole numbers as your test case.

**TAKE A SCREENSHOT** of the output console and paste it here. Do not paste your source code in this document.



1. [https://en.wikipedia.org/wiki/Methods\_of\_computing\_square\_roots - Babylonian\_method](https://en.wikipedia.org/wiki/Methods_of_computing_square_roots#Babylonian_method) [↑](#footnote-ref-1)