Programming Assignment Unit 5

Godfrey Ouma

University of the People

CS 1101: Programming Fundamentals

Janice Block

July 19, 2023

Programming Assignment Unit 5

Part 1: Encapsulating the code

Part 1 deals with encapsulating the code in the my_sqrt function as shown below.

```
In [185]: def my_sqrt(a):
     ...:     # Starting value for x
     ...:     x = 3.0
     ...:     while True:
     ...:         y = (x + a/x) / 2.0
     ...:         if y == x:
     ...:             break
     ...:         x = y
     ...:     return x
     ...:
     ...: my_sqrt(16)
Out[185]: 4.0
```

Explanation:

A while loop has been used in the "my_sqrt" function in the code above to execute the square root approximation using the s Newton's method. The function takes the parameter "a", which is the number to be substituted into the formula to estimate the square root. The starting value of "x" variable is 3.0. The loop keeps calculating the new estimate y and stops when y is equivalent to x to signify that the estimate has reached it stability.

Part 2: Writing a function named test_sqrt

Part 2 deals with writing the "test_sqrt" function to print the table as shown below

```
In [187]: import math
     ...:
     ...: def test_sqrt():
     ...:     print("{:<5} | {:<15} | {:<15} | {:<15}".format("a", "my_sqrt(a)", "math.sqrt(a)",
"diff"))
     ...:     print("-" * 60)
     ...:     for a in range(1, 26):
     ...:         my_sqrt_result = my_sqrt(a)
     ...:         math_sqrt_result = math.sqrt(a)
     ...:         diff = abs(my_sqrt_result - math_sqrt_result)
     ...:         print("{:<5} | {:<15} | {:<15} | {:<15}".format(a, my_sqrt_result,
math_sqrt_result, diff))
```

```
     ...: #Output
a     | my_sqrt(a)       | math.sqrt(a)     | diff
-------------------------------------------------------------
1     | 1.0              | 1.0              | 0.0
2     | 1.414213562373095 | 1.4142135623730951 | 2.220446049250313e-16
3     | 1.7320508075688772 | 1.7320508075688772 | 0.0
4     | 2.0              | 2.0              | 0.0
5     | 2.23606797749979 | 2.23606797749979 | 0.0
6     | 2.449489742783178 | 2.449489742783178 | 0.0
7     | 2.6457513110645907 | 2.6457513110645907 | 0.0
8     | 2.82842712474619 | 2.8284271247461903 | 4.440892098500626e-16
9     | 3.0              | 3.0              | 0.0
10    | 3.162277660168379 | 3.1622776601683795 | 4.440892098500626e-16
11    | 3.3166247903554 | 3.3166247903554 | 0.0
12    | 3.4641016151377544 | 3.4641016151377544 | 0.0
13    | 3.6055512754639896 | 3.605551275463989 | 4.440892098500626e-16
14    | 3.7416573867739413 | 3.7416573867739413 | 0.0
15    | 3.872983346207417 | 3.872983346207417 | 0.0
16    | 4.0              | 4.0              | 0.0
17    | 4.123105625617661 | 4.123105625617661 | 0.0
18    | 4.242640687119286 | 4.242640687119285 | 8.881784197001252e-16
19    | 4.358898943540673 | 4.358898943540674 | 8.881784197001252e-16
20    | 4.47213595499958 | 4.47213595499958 | 0.0
21    | 4.58257569495584 | 4.58257569495584 | 0.0
22    | 4.69041575982343 | 4.69041575982343 | 0.0
23    | 4.795831523312719 | 4.795831523312719 | 0.0
24    | 4.898979485566356 | 4.898979485566356 | 0.0
25    | 5.0              | 5.0              | 0.0
```

A for loop is used in the “test_sqrt” function to iterate over a values from 1 to 25. For each “a”, the my_sqrt function is called to get the estimated square root (my_sqrt_result) and calculate the square root using math.sqrt (math_sqrt_result). The difference (diff) between the two results is then computed using the absolute value (abs). Finally, the results are printed in a formatted table, as shown above.

Reference

Downey, A. (2015). *Think Python: How to think like a computer scientist*. Needham, Massachusetts: Green Tree