**Lecture 4.2. Functions**

**A picture containing table

Description automatically generated**

**Question 1.** For the code block on the right, write below the print messages in the order in which they appear when the code block is executed.

|  |  |
| --- | --- |
| Order | Print statement |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |

**Pro tip:** If you ever want to see line numbers for your code cell in Jupyter Notebook, go to *View* in the toolbar and select *Toggle Line Numbers*.

Graphical user interface, application

Description automatically generated

**Graphical user interface, text, application

Description automatically generatedQuestion 2.** Starting from line 10 in the code block on the right, fill in the table below tracking:

1. Flow of execution using the first *Line number* column
2. State of variables, for each line number

If the variable is (or becomes) undefined at any line of code, state so in the appropriate row of the table below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Line number** | **number** | **float\_number** | **fractional\_part** | **integer\_part** | **to\_ceil** | **rounded** | **number\_rounded** |
| 10 | 2.9 | undefined | undefined | undefined | undefined | undefined | undefined |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Question 3.** Implement a function square\_root that computes square root of the input. At the bottom are examples of how the function is expected to be called and the output(s) expected for given input(s).

num = 4

num\_sqrt = square\_root(num)

print(”num\_sqrt”) # Should print 2

num = 9

num\_sqrt = square\_root(num)

print(num\_sqrt) # Should print 3

num = 16

num\_sqrt = square\_root(num)

print(num\_sqrt) # Should print 4

**Question 4.** Implement a function euclidean that accepts four integers x1, y1, x2, y2 as inputs, representing two points (x1, y1) and (x2, y2). **Use square\_root from Question 3**.

x1 = 1

y1 = 5

x2 = 1

y2 = 5

dist = euclidean(x1, y1, x2, y2)

print(dist) # Should print 0

dist = euclidean(0, 25, 0, 16)

print(dist) # Should print 9