

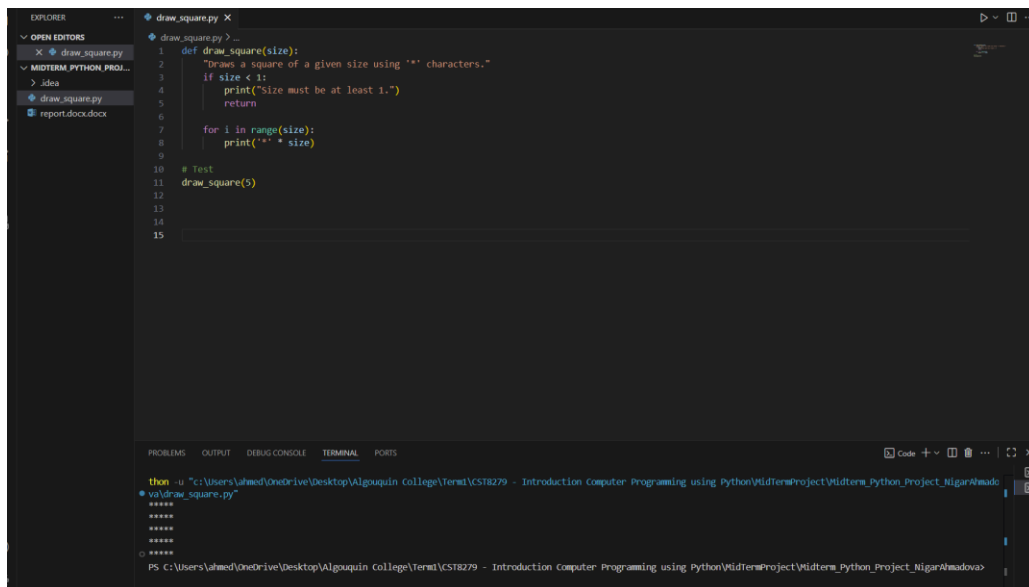
# Pattern Drawing in Python using Functions and OOP

## Project Description

This project creates a customizable square pattern using both a standalone function and an object-oriented class structure. It includes input validation, loop-based pattern generation, and a modular design that allows easy extension to other shapes. The project demonstrates proficiency in procedural programming, OOP concepts, and clean code practices.

## Function-Based Version

The first part of the project defines a function `draw_square(size)` that prints a square of `*` characters based on the given size. The function includes a docstring describing its purpose and checks if the provided size is less than 1, displaying an error message if so. If the input is valid, a for loop runs for the specified number of rows, printing a line of `*` characters equal to the given size.

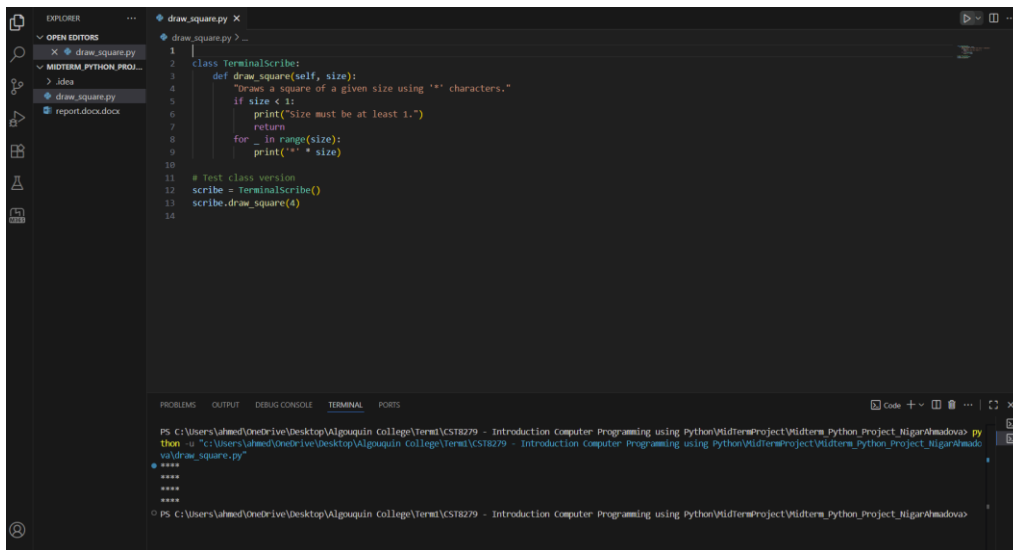


```
1 def draw_square(size):
2     """Draws a square of a given size using '*' characters."""
3     if size < 1:
4         print("Size must be at least 1.")
5         return
6
7     for i in range(size):
8         print('*' * size)
9
10 # Test
11 draw_square(5)
12
13
14
15
```

```
thon -u "C:\Users\ahmed\OneDrive\Desktop\Algaquin College\Term1\CS18279 - Introduction Computer Programming using Python\MidtermProject\Midterm_Python_Project_NigarAhmedoaz"
* valdraw_square.py"
*****
*****
*****
*****
*****
PS C:\Users\ahmed\OneDrive\Desktop\Algaquin College\Term1\CS18279 - Introduction Computer Programming using Python\MidtermProject\Midterm_Python_Project_NigarAhmedoaz>
```

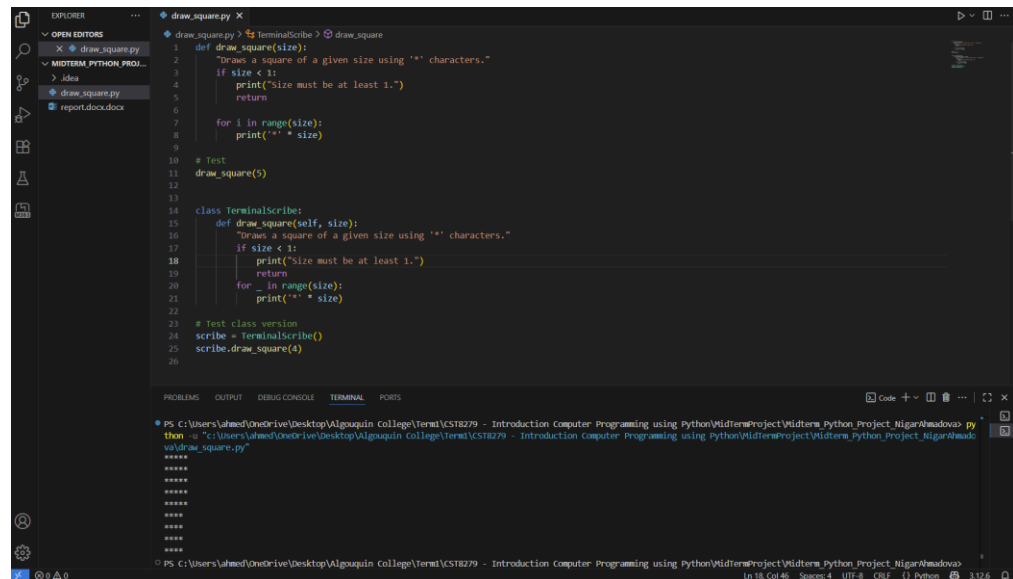
## Class-Based Version

The second implementation uses a class called `TerminalScribe`, which contains the `draw_square(self, size)` method. Like the function version, it includes a docstring, input validation, and the same loop logic for drawing the square. The method can be reused and extended to draw other shapes by adding more methods to the class.



```
1 class TerminalScribe:
2     def draw_square(self, size):
3         "Draws a square of a given size using '*' characters."
4         if size < 1:
5             print("Size must be at least 1.")
6             return
7         for _ in range(size):
8             print('*' * size)
9
10 # Test class version
11 scribe = TerminalScribe()
12 scribe.draw_square(4)
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
PS C:\Users\ahmed\OneDrive\Desktop\Algoquin College\Term1\CS18279 - Introduction Computer Programming using Python\MidtermProject\Midterm_Python_Project_NigarAhmadova> py
thon -u "C:\Users\ahmed\OneDrive\Desktop\Algoquin College\Term1\CS18279 - Introduction Computer Programming using Python\MidtermProject\Midterm_Python_Project_NigarAhmadova\draw_square.py"
****
****
****
****
PS C:\Users\ahmed\OneDrive\Desktop\Algoquin College\Term1\CS18279 - Introduction Computer Programming using Python\MidtermProject\Midterm_Python_Project_NigarAhmadova>
```



```
1 def draw_square(size):
2     "Draws a square of a given size using '*' characters."
3     if size < 1:
4         print("Size must be at least 1.")
5         return
6     for i in range(size):
7         print('*' * size)
8
9
10 # Test
11 draw_square(5)
12
13
14 class TerminalScribe:
15     def draw_square(self, size):
16         "Draws a square of a given size using '*' characters."
17         if size < 1:
18             print("Size must be at least 1.")
19             return
20         for _ in range(size):
21             print('*' * size)
22
23 # Test class version
24 scribe = TerminalScribe()
25 scribe.draw_square(4)
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
PS C:\Users\ahmed\OneDrive\Desktop\Algoquin College\Term1\CS18279 - Introduction Computer Programming using Python\MidtermProject\Midterm_Python_Project_NigarAhmadova> py
thon -u "C:\Users\ahmed\OneDrive\Desktop\Algoquin College\Term1\CS18279 - Introduction Computer Programming using Python\MidtermProject\Midterm_Python_Project_NigarAhmadova\draw_square.py"
*****
*****
*****
*****
*****
PS C:\Users\ahmed\OneDrive\Desktop\Algoquin College\Term1\CS18279 - Introduction Computer Programming using Python\MidtermProject\Midterm_Python_Project_NigarAhmadova>
```

## Shape Extensions

The TerminalScribe class could be expanded with additional methods to draw other shapes, such as rectangles, triangles, or hollow squares. These shapes would require adjusting loop logic and conditional statements, making the design scalable and easy to extend.