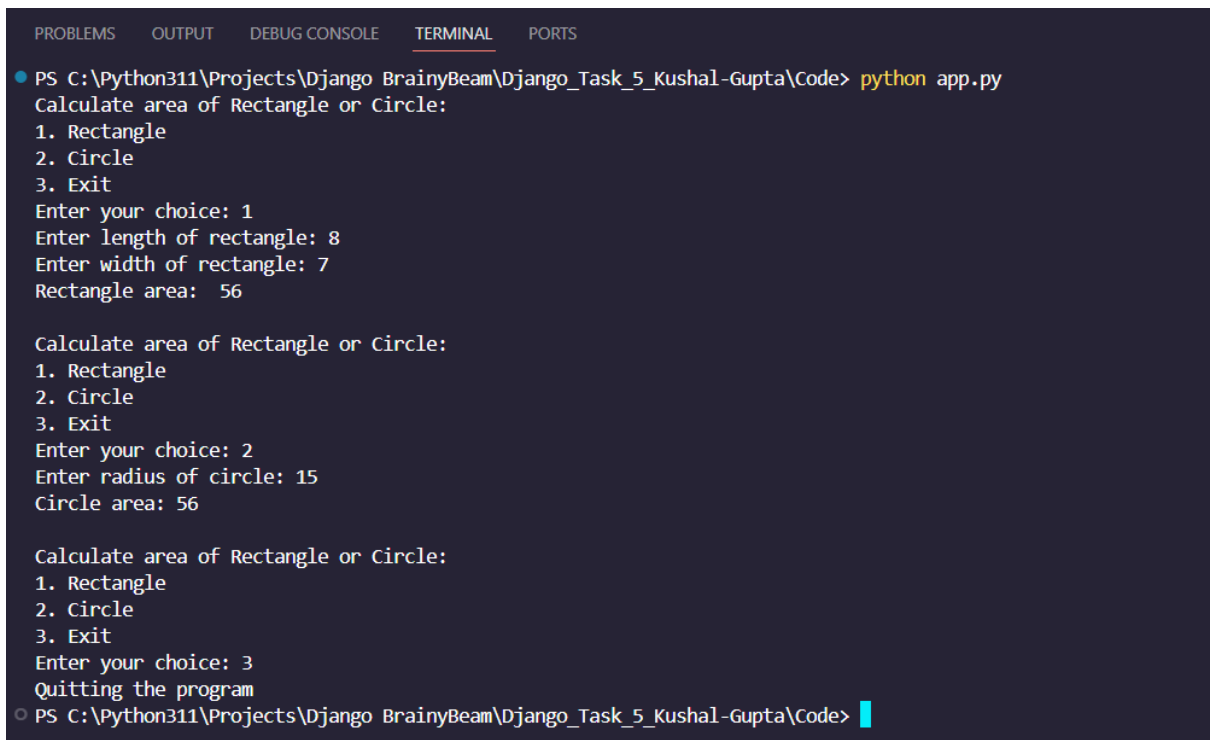


Task 5

1. Task Description:

Build a python program which can display advantages of the polymorphism and encapsulation in same program.

2. Task Output Screenshot:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Python311\Projects\Django BrainyBeam\Django_Task_5_Kushal-Gupta\Code> python app.py
Calculate area of Rectangle or Circle:
1. Rectangle
2. Circle
3. Exit
Enter your choice: 1
Enter length of rectangle: 8
Enter width of rectangle: 7
Rectangle area: 56

Calculate area of Rectangle or Circle:
1. Rectangle
2. Circle
3. Exit
Enter your choice: 2
Enter radius of circle: 15
Circle area: 56

Calculate area of Rectangle or Circle:
1. Rectangle
2. Circle
3. Exit
Enter your choice: 3
Quitting the program
PS C:\Python311\Projects\Django BrainyBeam\Django_Task_5_Kushal-Gupta\Code>
```

3. Algorithm Used in Task:

The algorithm implements an Object-Oriented Shape Calculator that uses polymorphism and encapsulation to handle different geometric shapes. The program provides a menu-driven interface allowing users to calculate areas of different shapes while maintaining data integrity and code reusability.

- 1) Encapsulation Implementation:
 - a. Private attributes for dimensions (`__length`, `__width`, `__radius`)
 - b. Public setter methods with validation logic
 - c. Protected access to internal data through controlled methods
- 2) Polymorphic Structure:
 - a. Different shapes inherit from base Shape class

- b. Each shape provides its own area calculation logic
- c. Common interface allows uniform handling of different shapes

This algorithm provides a robust and extensible way to handle geometric calculations while maintaining object-oriented principles. It demonstrates how encapsulation and polymorphism can be used to create a maintainable and scalable program structure that can be easily extended to include additional shapes.