**Python Assignment 2**

Objective:

The objective of this assignment is to develop Python programming skills with a focus on functions

and object-oriented programming (OOP). Participants will learn to:

• Define and use functions efficiently.

• Work with function arguments, including \*args and \*\*kwargs.

• Utilize lambda functions and built-in higher-order functions.

• Implement OOP concepts such as classes, objects, inheritance, and polymorphism.

This assignment will enhance problem-solving abilities, promote code reusability, and introduce

participants to real-world programming patterns.

Functions

**1. Defining and Calling Functions**

•1a.Write a function calculate\_area() that takes length and width as arguments and returns the

area of a rectangle.

•1b.Write a function that takes a list of numbers and returns the sum of even numbers only.

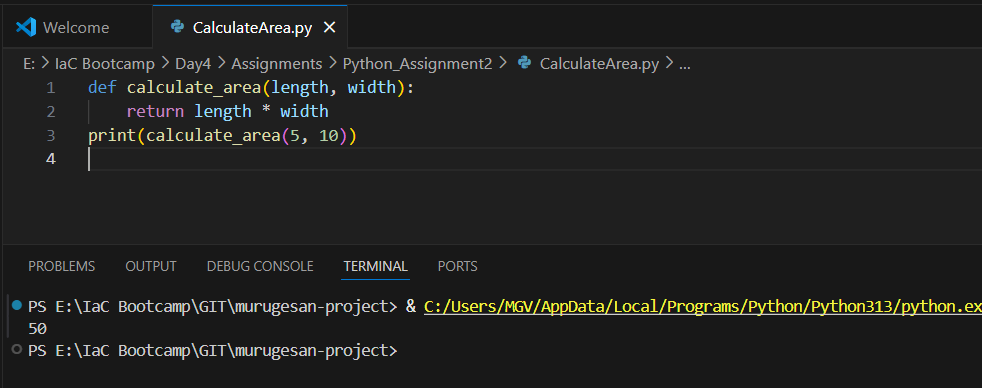
Sample Output:

# calculate\_area()

print(calculate\_area(5, 10))

# Output: 50

**Screenshot:**

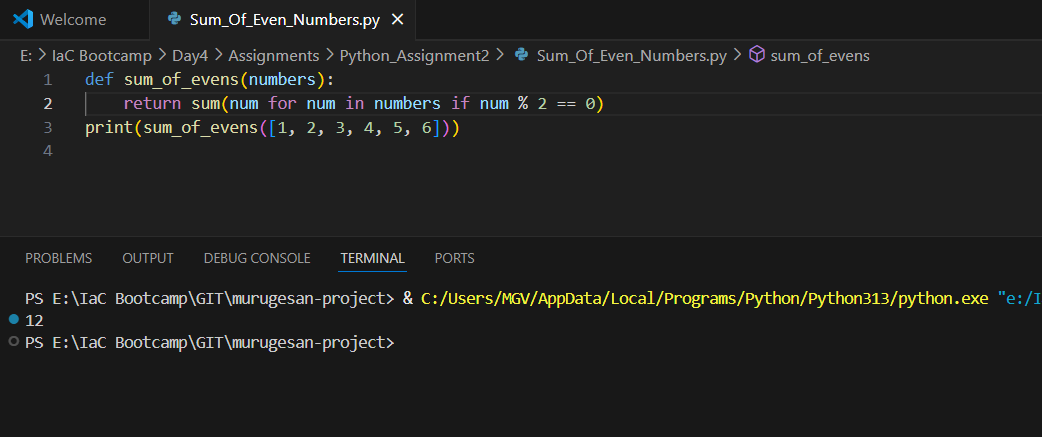


# sum\_of\_evens()

print(sum\_of\_evens([1, 2, 3, 4, 5, 6]))

# Output: 12

**Screenshot:**

****

**2. Function Arguments**

2a. Create a function describe\_person() that takes positional arguments (name, age) and keyword

arguments (city, profession), then prints a formatted introduction.

2b.Implement a function that takes a variable number of arguments using \*args and \*\*kwargs,

then prints them.

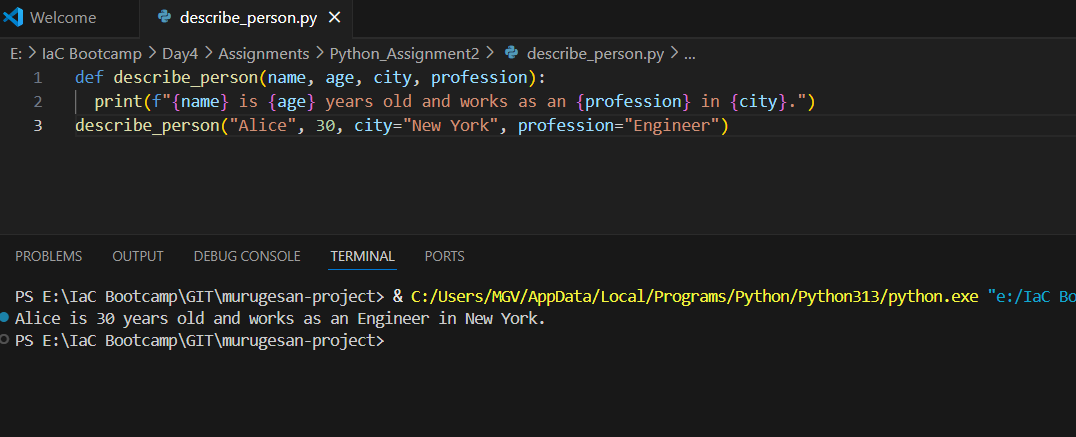
Sample Output:

describe\_person("Alice", 30, city="New York", profession="Engineer")

# Output: Alice is 30 years old and works as an Engineer in New York.

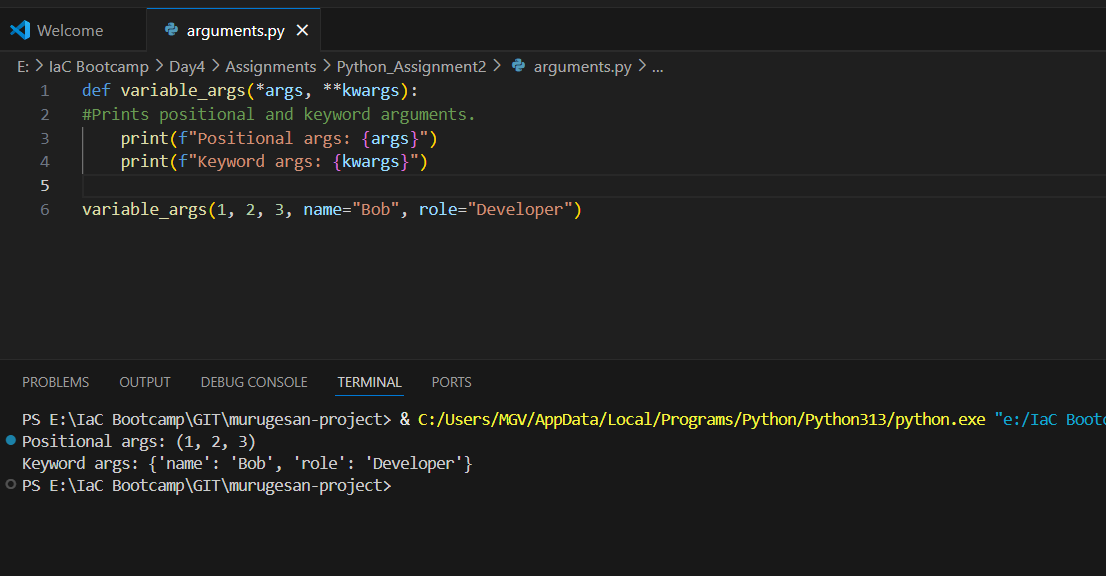
**variable\_args(1, 2, 3, name="Bob", role="Developer")**

**Screenshot:**

****

**# Output: Positional args: (1, 2, 3)**

**# Keyword args: {'name': 'Bob', 'role': 'Developer'}**

****

**Object-Oriented Programming (OOP) in Python**

3. Classes and Objects

3a. Create a BankAccount class with attributes account\_holder, balance, and methods to deposit

and withdraw money.

3b. Implement a Book class that holds book details (title, author, price). Create instances and print

book details.

Sample Output:

account = BankAccount("John Doe", 1000)

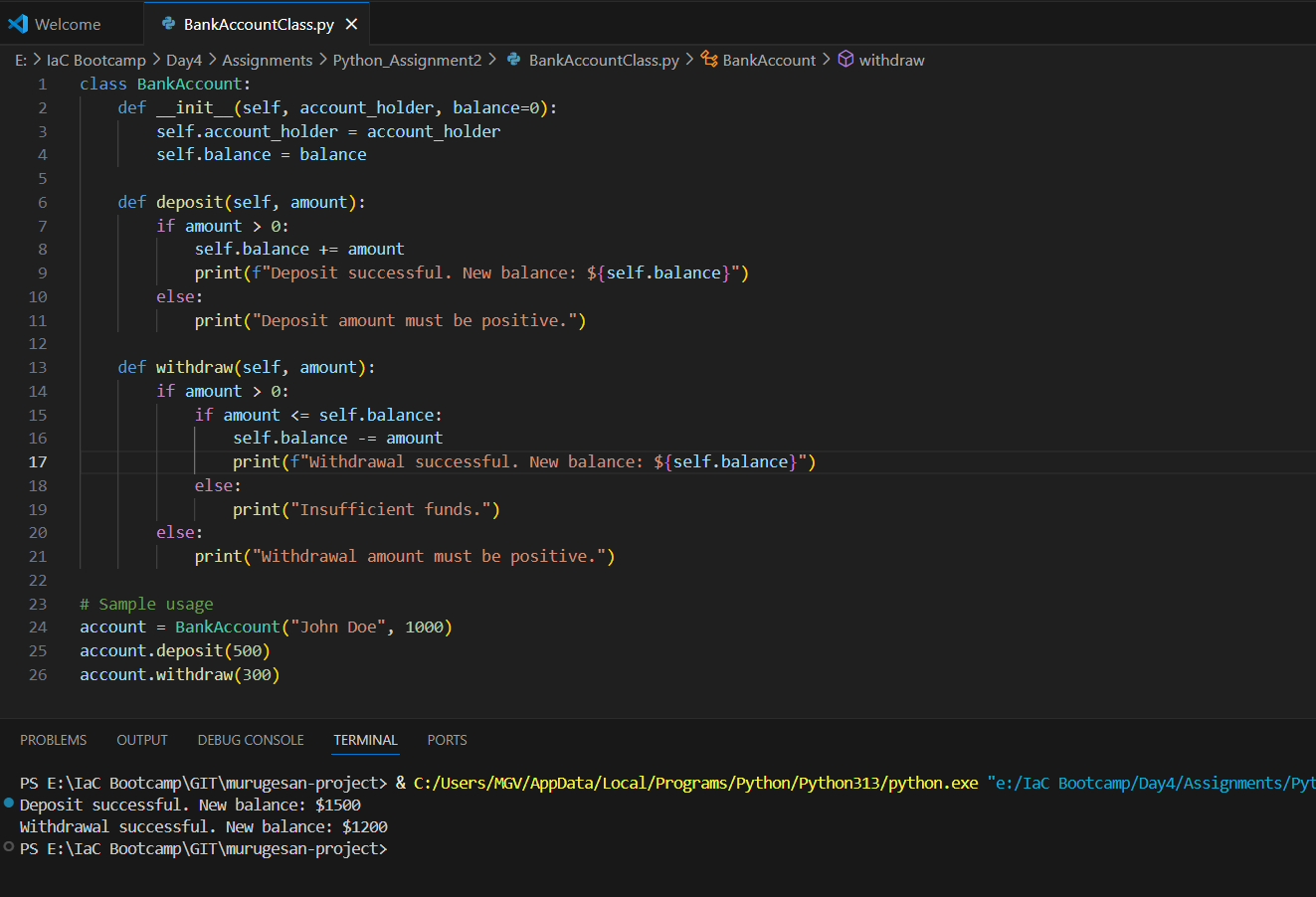
account.deposit(500)

account.withdraw(300)

# Output: Deposit successful. New balance: $1500

# Withdrawal successful. New balance: $1200

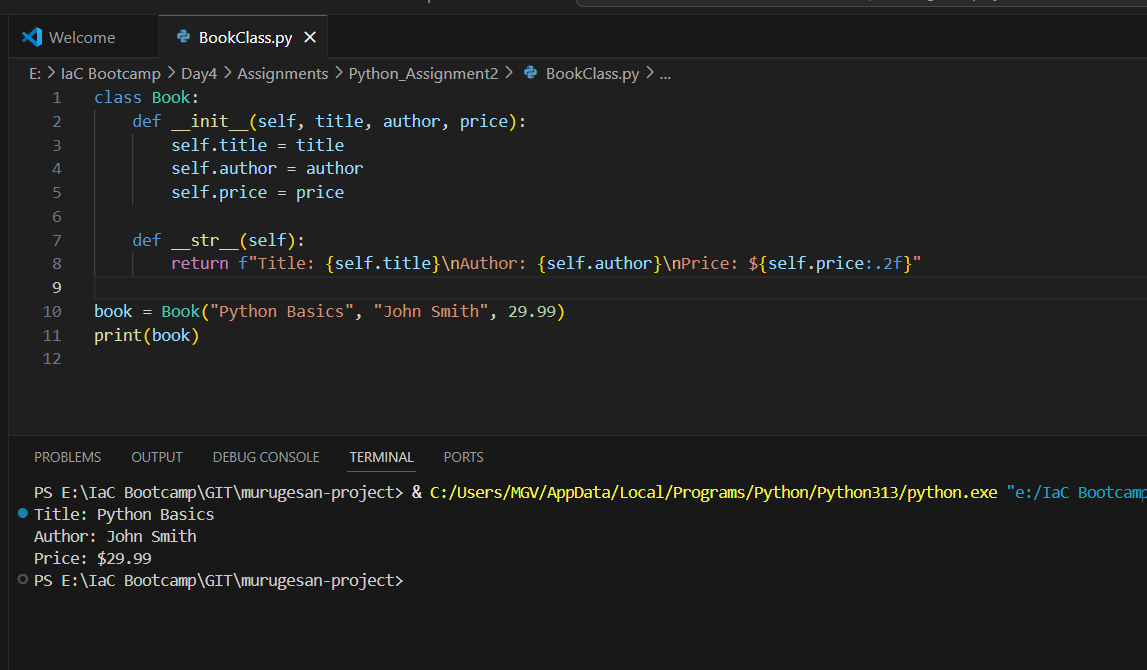
**Screenshot:**

****

book = Book("Python Basics", "John Smith", 29.99)

print(book)

# Output: Title: Python Basics, Author: John Smith, Price: $29.99



4. Constructors (\_\_init\_\_ Method)

• Implement a Student class where attributes (name, age, grade) are initialized in the

constructor.

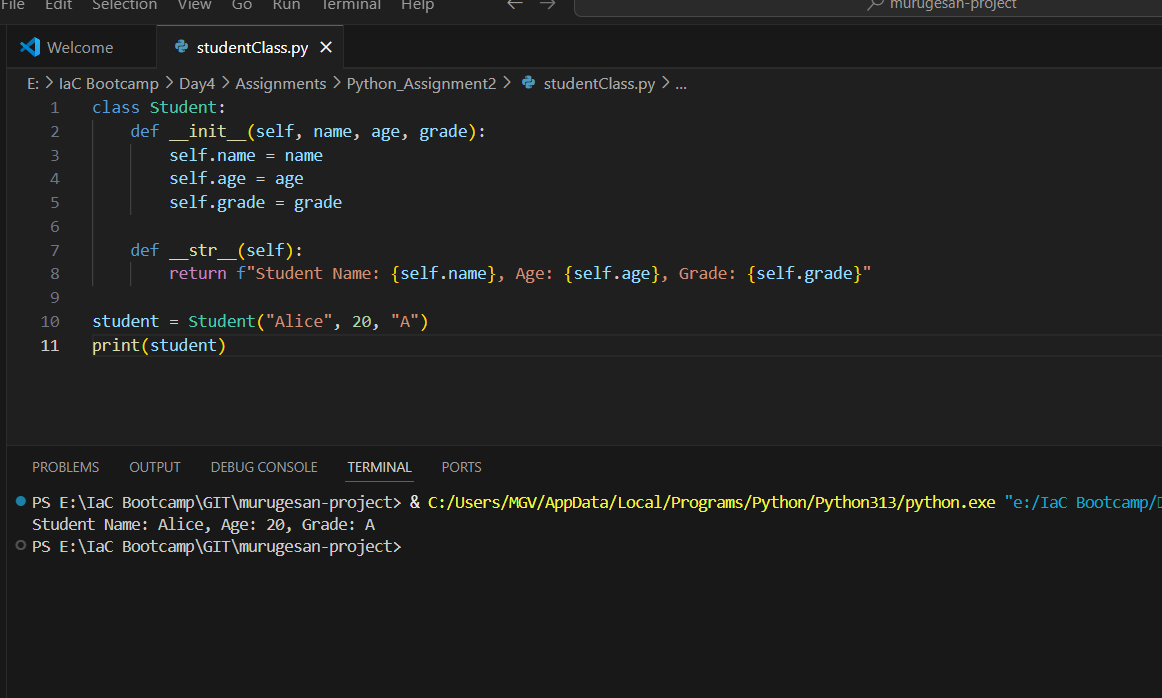
Sample Output:

student = Student("Alice", 20, "A")

print(student)

# Output: Student Name: Alice, Age: 20, Grade: A

**Screenshot:**



5. Instance and Class Variables

• Create a Car class with an instance variable color and a class variable wheels set to 4.

Sample Output:

car1 = Car("Red")

car2 = Car("Blue")

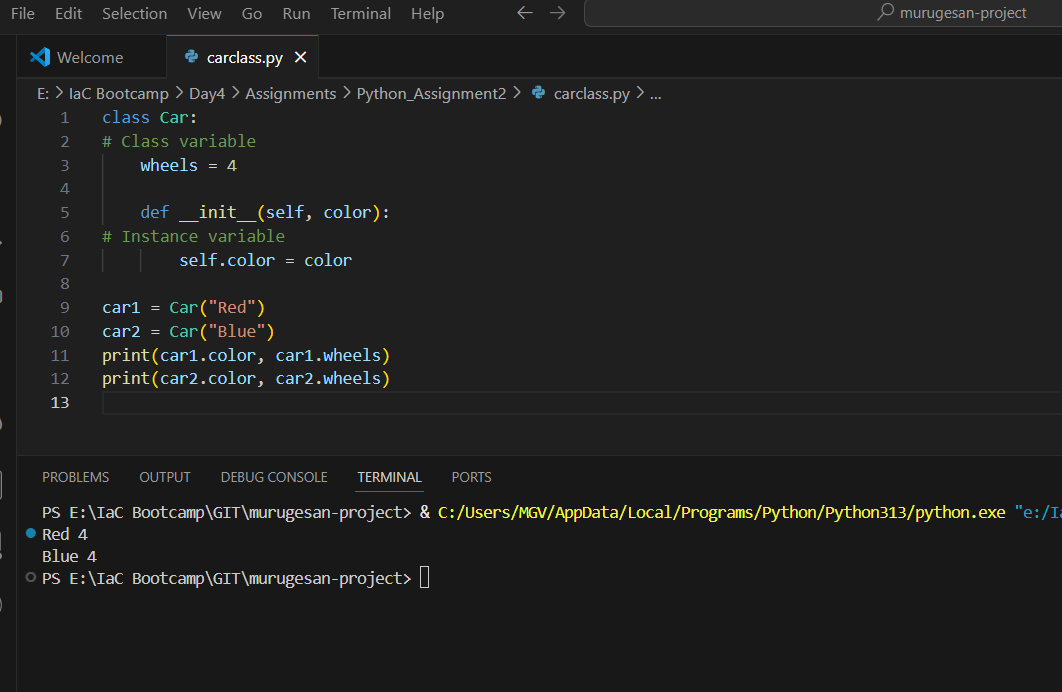
print(car1.color, car1.wheels)

# Output: Red 4

print(car2.color, car2.wheels)

# Output: Blue 4

**Screenshot:**



Assignment Suggestions:

✔ Use clear function and class names to improve readability.

✔ Follow PEP 8 guidelines for formatting and maintain proper indentation.

✔ Test each function/class thoroughly before submission.

✔ Comment your code to explain complex logic.

✔ Handle edge cases (e.g., invalid inputs for calculations).