AI Lab

Journal # 4

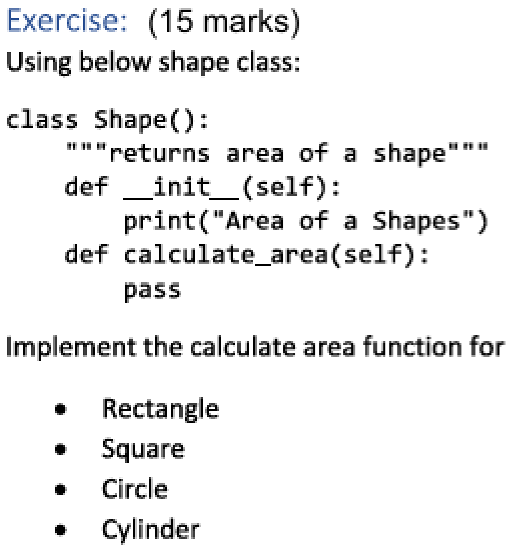
Talha Bin Tahir (01-131222-047)

Class: BSE-6A

Presented to: Engr. Aamir

Date: March 4, 2025

Polymorphism and Inheritance



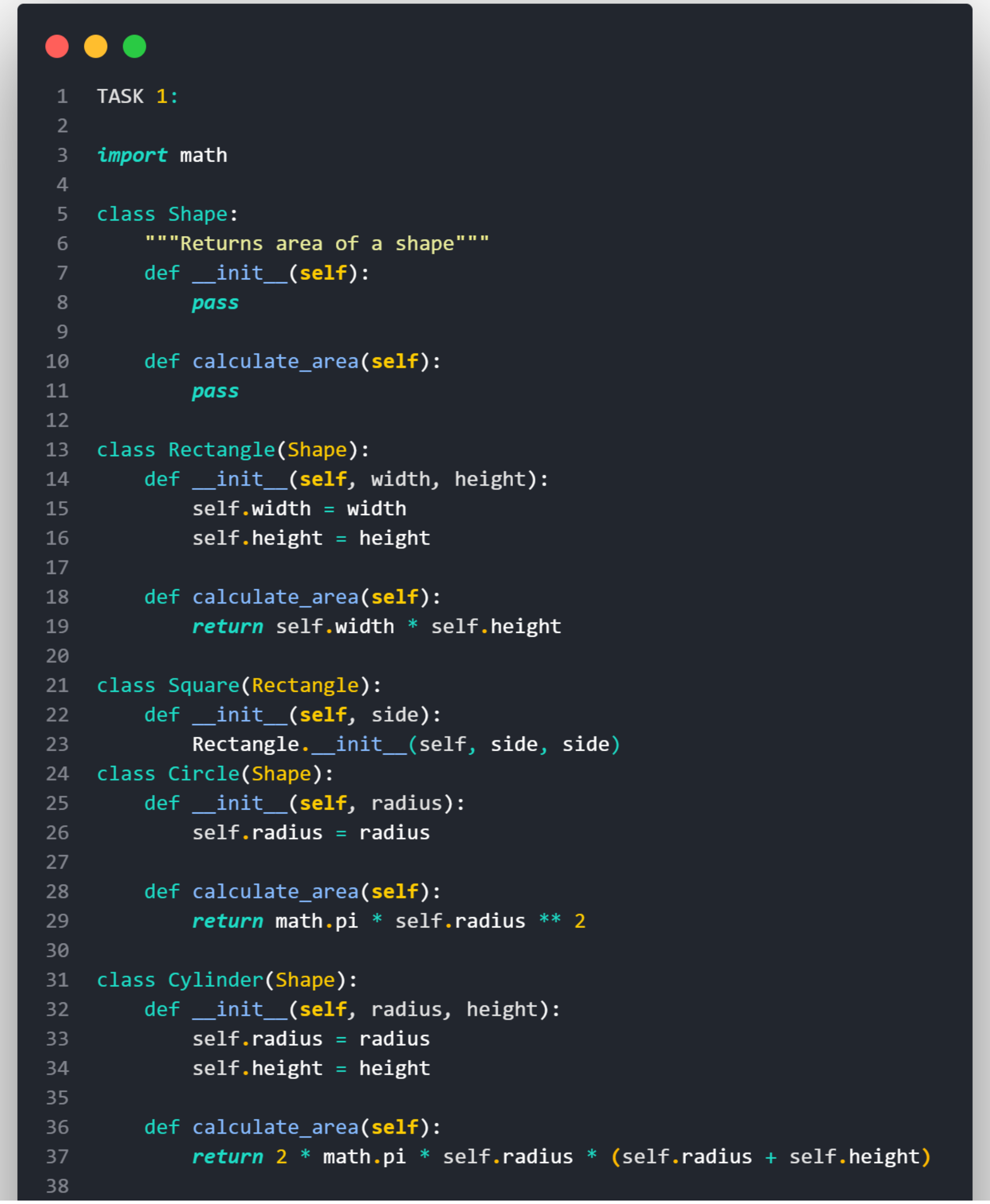
Exercise: (15 marks)

Using below shape class:

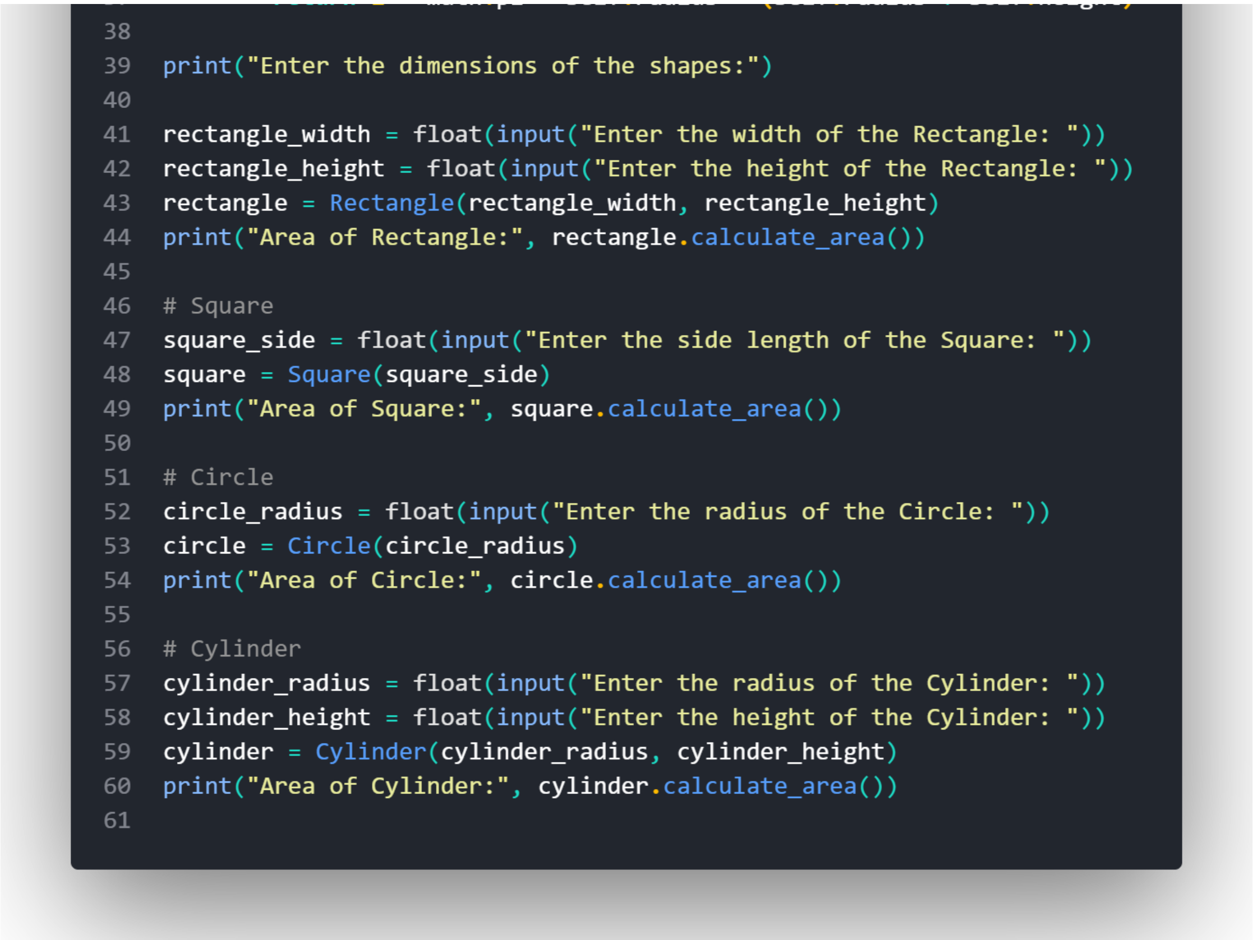
```
class Shape():
    """returns area of a shape"""
    def __init__(self):
        print("Area of a Shapes")
    def calculate_area(self):
        pass
```

Implement the calculate area function for

- Rectangle
- Square
- Circle
- Cylinder

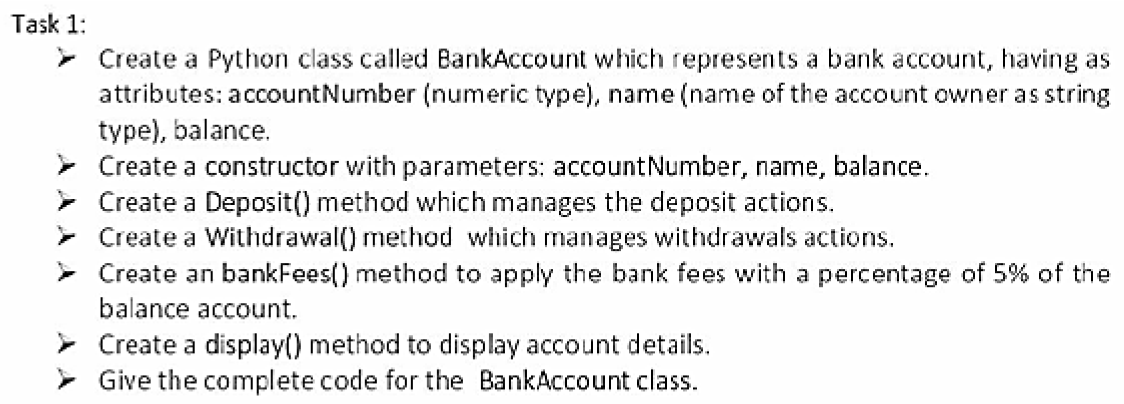


```
TASK 1:

import math

class Shape:
    """Returns area of a shape"""
    def __init__(self):
        pass

    def calculate_area(self):
        pass

class Rectangle(Shape):
    def __init__(self, width, height):
        self.width = width
        self.height = height

    def calculate_area(self):
        return self.width * self.height

class Square(Rectangle):
    def __init__(self, side):
        Rectangle.__init__(self, side, side)
class Circle(Shape):
    def __init__(self, radius):
        self.radius = radius

    def calculate_area(self):
        return math.pi * self.radius ** 2

class Cylinder(Shape):
    def __init__(self, radius, height):
        self.radius = radius
        self.height = height

    def calculate_area(self):
        return 2 * math.pi * self.radius * (self.radius + self.height)

```



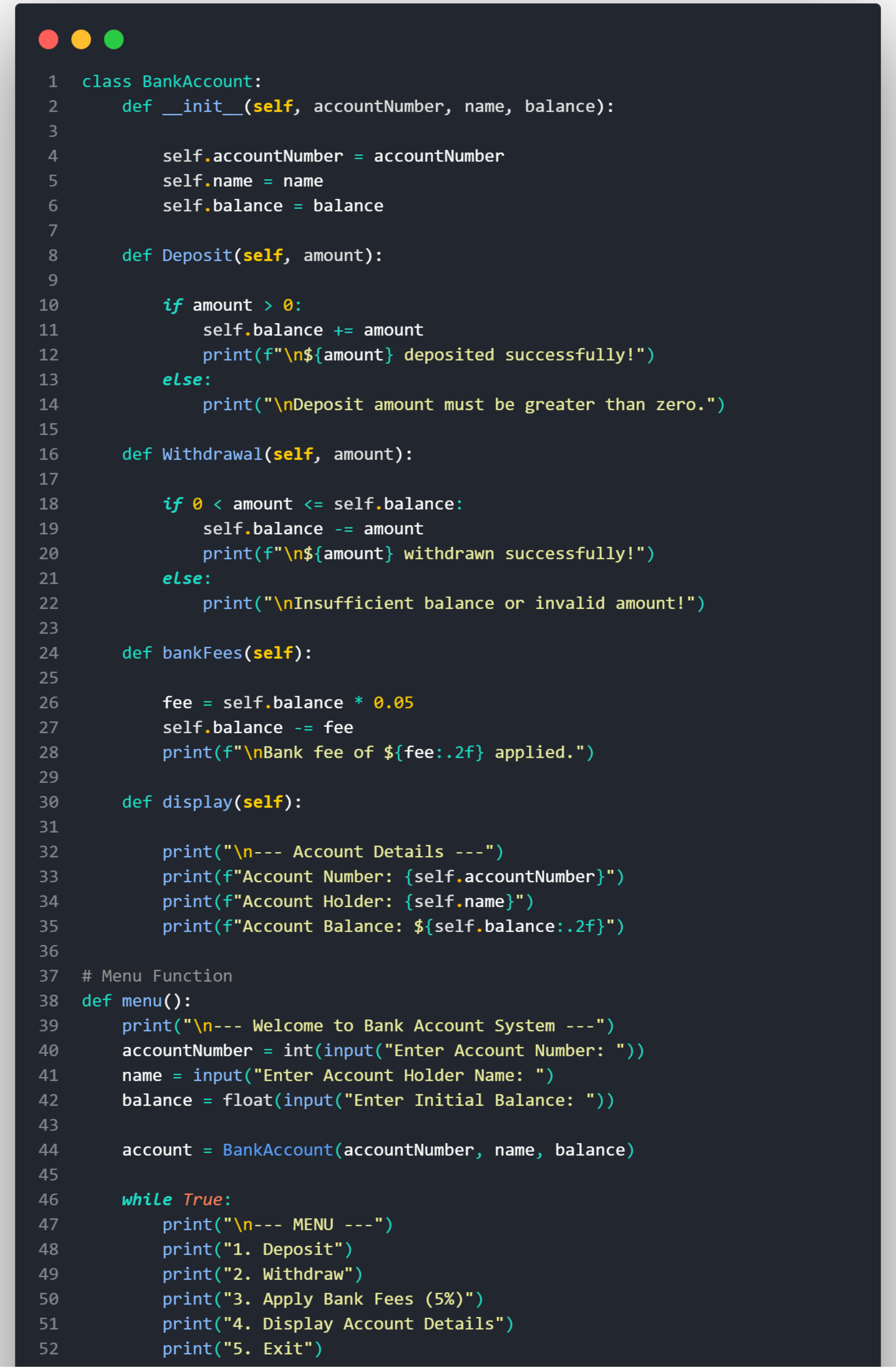
```
        return 2 * math.pi * self.radius * (self.radius + self.height)

print("Enter the dimensions of the shapes:")

rectangle_width = float(input("Enter the width of the Rectangle: "))
rectangle_height = float(input("Enter the height of the Rectangle: "))
rectangle = Rectangle(rectangle_width, rectangle_height)
print("Area of Rectangle:", rectangle.calculate_area())

# Square
square_side = float(input("Enter the side length of the Square: "))
square = Square(square_side)
print("Area of Square:", square.calculate_area())

# Circle
circle_radius = float(input("Enter the radius of the Circle: "))
circle = Circle(circle_radius)
print("Area of Circle:", circle.calculate_area())

# Cylinder
cylinder_radius = float(input("Enter the radius of the Cylinder: "))
cylinder_height = float(input("Enter the height of the Cylinder: "))
cylinder = Cylinder(cylinder_radius, cylinder_height)
print("Area of Cylinder:", cylinder.calculate_area())

```

EXTRA TASKS:

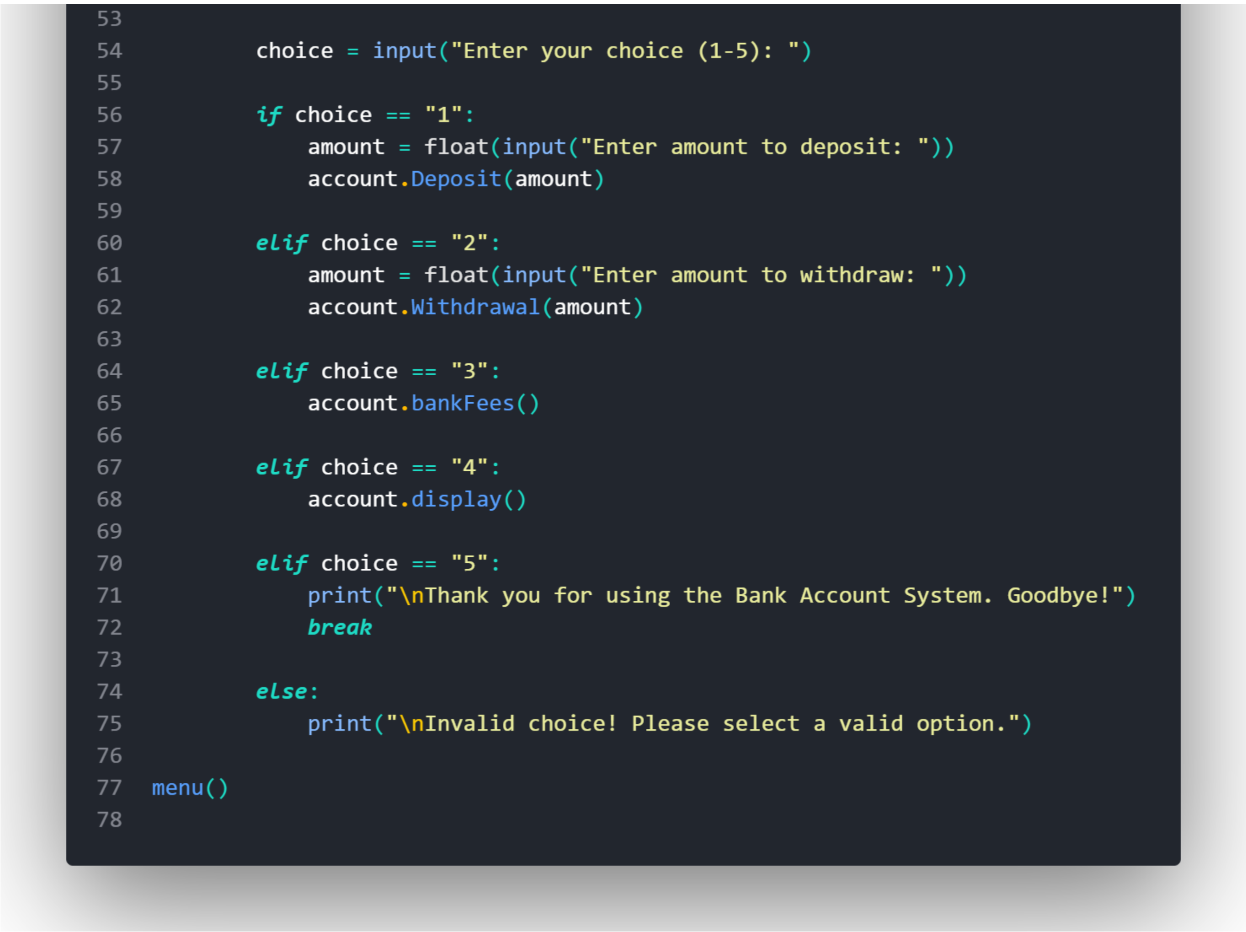


Task 1:

- Create a Python class called BankAccount which represents a bank account, having as attributes: accountNumber (numeric type), name (name of the account owner as string type), balance.
- Create a constructor with parameters: accountNumber, name, balance.
- Create a Deposit() method which manages the deposit actions.
- Create a Withdrawal() method which manages withdrawals actions.
- Create an bankFees() method to apply the bank fees with a percentage of 5% of the balance account.
- Create a display() method to display account details.
- Give the complete code for the BankAccount class.

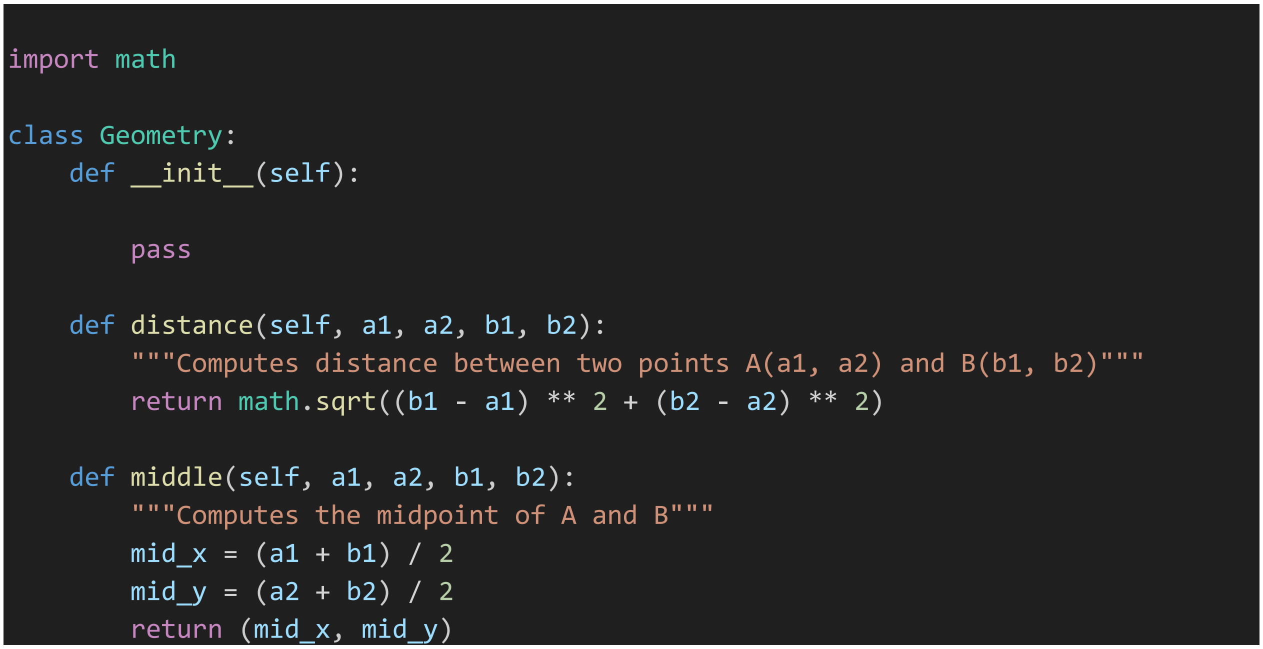


```python
class BankAccount:
    def __init__(self, accountNumber, name, balance):

        self.accountNumber = accountNumber
        self.name = name
        self.balance = balance

    def Deposit(self, amount):

        if amount > 0:
            self.balance += amount
            print(f"\n${amount} deposited successfully!")
        else:
            print("\nDeposit amount must be greater than zero.")

    def Withdrawal(self, amount):

        if 0 < amount <= self.balance:
            self.balance -= amount
            print(f"\n${amount} withdrawn successfully!")
        else:
            print("\nInsufficient balance or invalid amount!")

    def bankFees(self):

        fee = self.balance * 0.05
        self.balance -= fee
        print(f"\nBank fee of ${fee:.2f} applied.")

    def display(self):

        print("\n--- Account Details ---")
        print(f"Account Number: {self.accountNumber}")
        print(f"Account Holder: {self.name}")
        print(f"Account Balance: ${self.balance:.2f}")

# Menu Function
def menu():
    print("\n--- Welcome to Bank Account System ---")
    accountNumber = int(input("Enter Account Number: "))
    name = input("Enter Account Holder Name: ")
    balance = float(input("Enter Initial Balance: "))

    account = BankAccount(accountNumber, name, balance)

    while True:
        print("\n--- MENU ---")
        print("1. Deposit")
        print("2. Withdraw")
        print("3. Apply Bank Fees (5%)")
        print("4. Display Account Details")
        print("5. Exit")
```

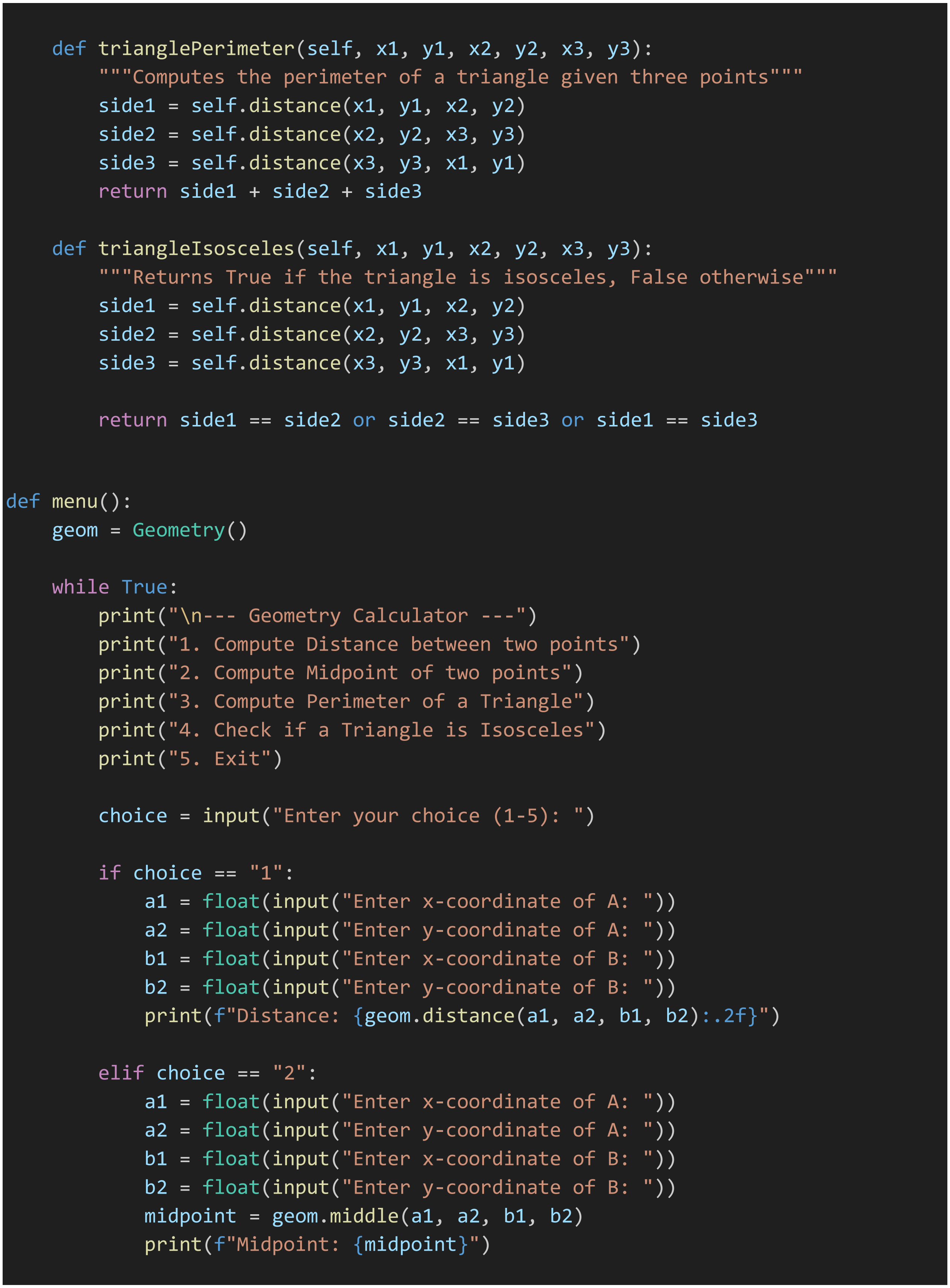


```python

        choice = input("Enter your choice (1-5): ")

        if choice == "1":
            amount = float(input("Enter amount to deposit: "))
            account.Deposit(amount)

        elif choice == "2":
            amount = float(input("Enter amount to withdraw: "))
            account.Withdrawal(amount)

        elif choice == "3":
            account.bankFees()

        elif choice == "4":
            account.display()

        elif choice == "5":
            print("\nThank you for using the Bank Account System. Goodbye!")
            break

        else:
            print("\nInvalid choice! Please select a valid option.")

menu()

```

TASK 3:



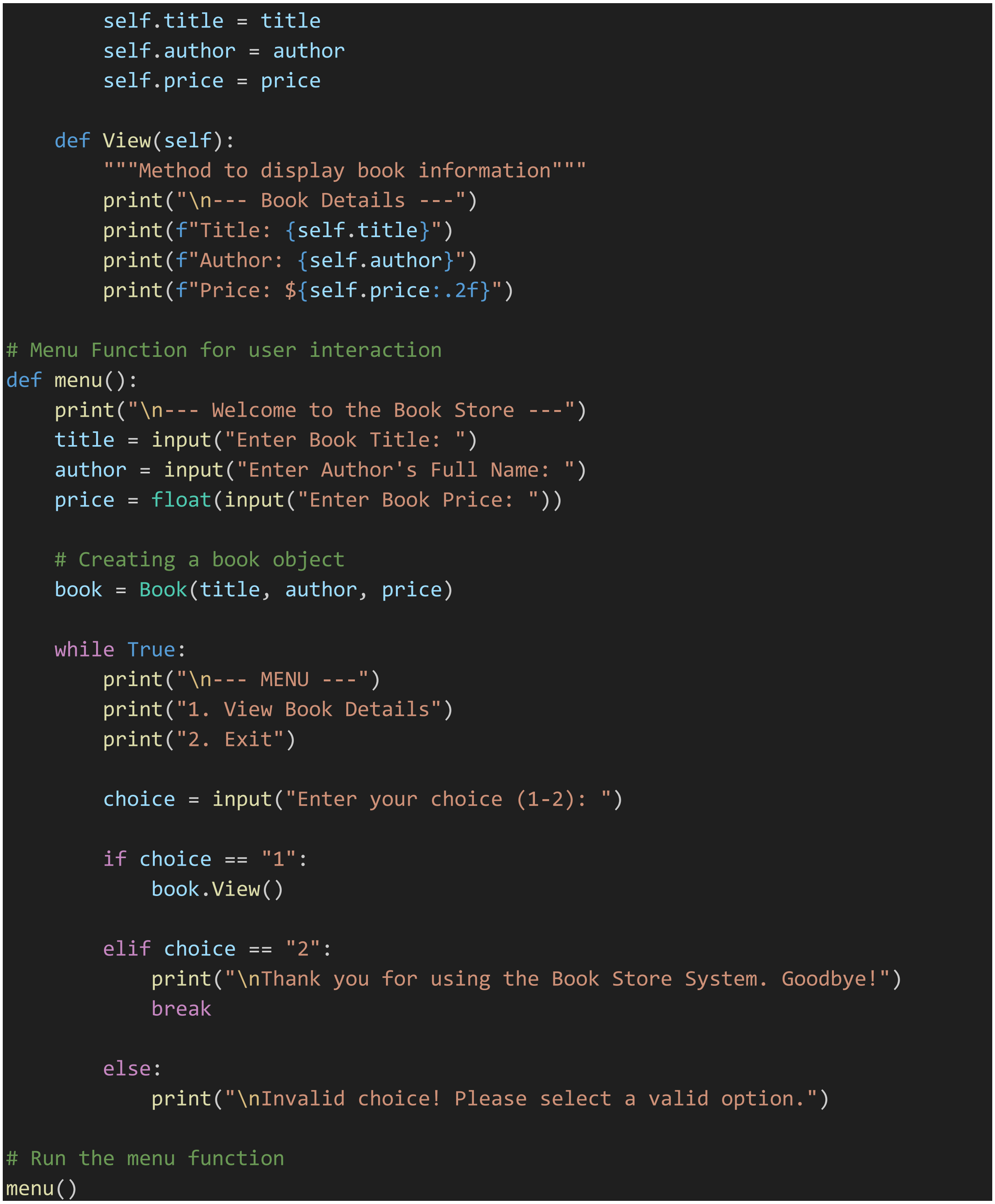
```python
import math

class Geometry:
    def __init__(self):

        pass

    def distance(self, a1, a2, b1, b2):
        """Computes distance between two points A(a1, a2) and B(b1, b2)"""
        return math.sqrt((b1 - a1) ** 2 + (b2 - a2) ** 2)

    def middle(self, a1, a2, b1, b2):
        """Computes the midpoint of A and B"""
        mid_x = (a1 + b1) / 2
        mid_y = (a2 + b2) / 2
        return (mid_x, mid_y)
```



```python
    def trianglePerimeter(self, x1, y1, x2, y2, x3, y3):
        """Computes the perimeter of a triangle given three points"""
        side1 = self.distance(x1, y1, x2, y2)
        side2 = self.distance(x2, y2, x3, y3)
        side3 = self.distance(x3, y3, x1, y1)
        return side1 + side2 + side3

    def triangleIsosceles(self, x1, y1, x2, y2, x3, y3):
        """Returns True if the triangle is isosceles, False otherwise"""
        side1 = self.distance(x1, y1, x2, y2)
        side2 = self.distance(x2, y2, x3, y3)
        side3 = self.distance(x3, y3, x1, y1)

        return side1 == side2 or side2 == side3 or side1 == side3


def menu():
    geom = Geometry()

    while True:
        print("\n--- Geometry Calculator ---")
        print("1. Compute Distance between two points")
        print("2. Compute Midpoint of two points")
        print("3. Compute Perimeter of a Triangle")
        print("4. Check if a Triangle is Isosceles")
        print("5. Exit")

        choice = input("Enter your choice (1-5): ")

        if choice == "1":
            a1 = float(input("Enter x-coordinate of A: "))
            a2 = float(input("Enter y-coordinate of A: "))
            b1 = float(input("Enter x-coordinate of B: "))
            b2 = float(input("Enter y-coordinate of B: "))
            print(f"Distance: {geom.distance(a1, a2, b1, b2):.2f}")

        elif choice == "2":
            a1 = float(input("Enter x-coordinate of A: "))
            a2 = float(input("Enter y-coordinate of A: "))
            b1 = float(input("Enter x-coordinate of B: "))
            b2 = float(input("Enter y-coordinate of B: "))
            midpoint = geom.middle(a1, a2, b1, b2)
            print(f"Midpoint: {midpoint}")
        elif choice == "3":
            x1 = float(input("Enter x-coordinate of Point 1: "))
            y1 = float(input("Enter y-coordinate of Point 1: "))
            x2 = float(input("Enter x-coordinate of Point 2: "))
            y2 = float(input("Enter y-coordinate of Point 2: "))
            x3 = float(input("Enter x-coordinate of Point 3: "))
            y3 = float(input("Enter y-coordinate of Point 3: "))
            perimeter = geom.trianglePerimeter(x1, y1, x2, y2, x3, y3)
            print(f"Triangle Perimeter: {perimeter:.2f}")

        elif choice == "4":
            x1 = float(input("Enter x-coordinate of Point 1: "))
            y1 = float(input("Enter y-coordinate of Point 1: "))
            x2 = float(input("Enter x-coordinate of Point 2: "))
            y2 = float(input("Enter y-coordinate of Point 2: "))
            x3 = float(input("Enter x-coordinate of Point 3: "))
            y3 = float(input("Enter y-coordinate of Point 3: "))
            is_isosceles = geom.triangleIsosceles(x1, y1, x2, y2, x3, y3)
            print("Triangle is Isosceles" if is_isosceles else "Triangle is NOT Isosceles")

        elif choice == "5":
            print("\nExiting... Thank you for using the Geometry Calculator!")
            break

        else:
            print("\nInvalid choice! Please select a valid option.")

menu()
```

TASK 4:

```
class Book:
    def __init__(self, title, author, price):
        """Constructor to initialize the book details"""
```



```
        self.title = title
        self.author = author
        self.price = price

    def View(self):
        """Method to display book information"""
        print("\n--- Book Details ---")
        print(f"Title: {self.title}")
        print(f"Author: {self.author}")
        print(f"Price: ${self.price:.2f}")

# Menu Function for user interaction
def menu():
    print("\n--- Welcome to the Book Store ---")
    title = input("Enter Book Title: ")
    author = input("Enter Author's Full Name: ")
    price = float(input("Enter Book Price: "))

    # Creating a book object
    book = Book(title, author, price)

    while True:
        print("\n--- MENU ---")
        print("1. View Book Details")
        print("2. Exit")

        choice = input("Enter your choice (1-2): ")

        if choice == "1":
            book.View()

        elif choice == "2":
            print("\nThank you for using the Book Store System. Goodbye!")
            break

        else:
            print("\nInvalid choice! Please select a valid option.")

# Run the menu function
menu()
```