LAB CYCLE 6

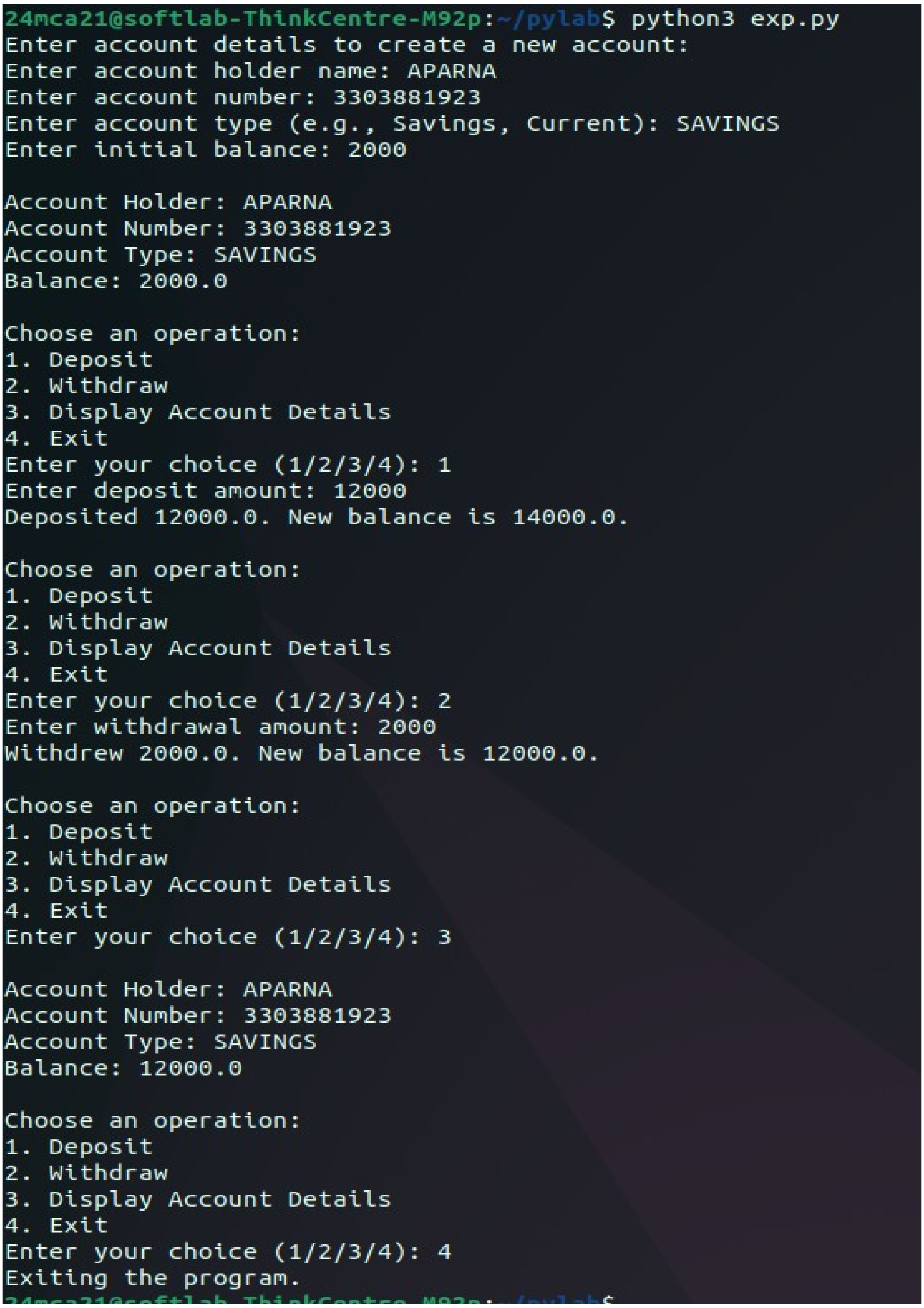
PROGRAM NO:1
AIM
Define a class to represent a bank account. Include the following details like name of the depositor, account number, type of account, balance amount in the account. Write methods to assign initial values, to deposit an amount, withdraw an amount after checking the balance, to display details such as name, account number, account type and balance.

SOURCE CODE

```
class BankAccount:
    def __init__(self, name, account_number, account_type, balance):
        self.name = name
        self.account_number = account_number
        self.account_type = account_type
        self.balance = balance

    def deposit(self, amount):
        """Method to deposit an amount into the account."""
        if amount > 0:
            self.balance += amount
            print(f"Deposited {amount}. New balance is {self.balance}.")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        """Method to withdraw an amount from the account."""
        if amount <= 0:
            print("Withdrawal amount must be positive.")
        elif amount > self.balance:
            print("Insufficient balance.")
        else:
            self.balance -= amount
            print(f"Withdrew {amount}. New balance is {self.balance}.")

    def display_details(self):
        """Method to display account details."""
        print(f"\nAccount Holder: {self.name}")
        print(f"Account Number: {self.account_number}")
        print(f"Account Type: {self.account_type}")
        print(f"Balance: {self.balance}")

print("Enter account details to create a new account:")
name = input("Enter account holder name: ")
account_number = input("Enter account number: ")
account_type = input("Enter account type (e.g., Savings, Current): ")
balance = float(input("Enter initial balance: "))

account1 = BankAccount(name, account_number, account_type, balance)
account1.display_details()
while True:
```

```
print("\nChoose an operation:")
print("1. Deposit")
print("2. Withdraw")
print("3. Display Account Details")
print("4. Exit")

choice = input("Enter your choice (1/2/3/4): ")

if choice == "1":
    deposit_amount = float(input("Enter deposit amount: "))
    account1.deposit(deposit_amount)
elif choice == "2":
    withdraw_amount = float(input("Enter withdrawal amount: "))
    account1.withdraw(withdraw_amount)
elif choice == "3":
    account1.display_details()
elif choice == "4":
    print("Exiting the program.")
    break
else:
    print("Invalid choice. Please try again.")
```

OUTPUT



```
24mca21@softlab-ThinkCentre-M92p:~/pylab$ python3 exp.py
Enter account details to create a new account:
Enter account holder name: APARNA
Enter account number: 3303881923
Enter account type (e.g., Savings, Current): SAVINGS
Enter initial balance: 2000

Account Holder: APARNA
Account Number: 3303881923
Account Type: SAVINGS
Balance: 2000.0

Choose an operation:
1. Deposit
2. Withdraw
3. Display Account Details
4. Exit
Enter your choice (1/2/3/4): 1
Enter deposit amount: 12000
Deposited 12000.0. New balance is 14000.0.

Choose an operation:
1. Deposit
2. Withdraw
3. Display Account Details
4. Exit
Enter your choice (1/2/3/4): 2
Enter withdrawal amount: 2000
Withdrew 2000.0. New balance is 12000.0.

Choose an operation:
1. Deposit
2. Withdraw
3. Display Account Details
4. Exit
Enter your choice (1/2/3/4): 3

Account Holder: APARNA
Account Number: 3303881923
Account Type: SAVINGS
Balance: 12000.0

Choose an operation:
1. Deposit
2. Withdraw
3. Display Account Details
4. Exit
Enter your choice (1/2/3/4): 4
Exiting the program.
24mca21@softlab-ThinkCentre-M92p:~/pylab$
```

PROGRAM NO:2

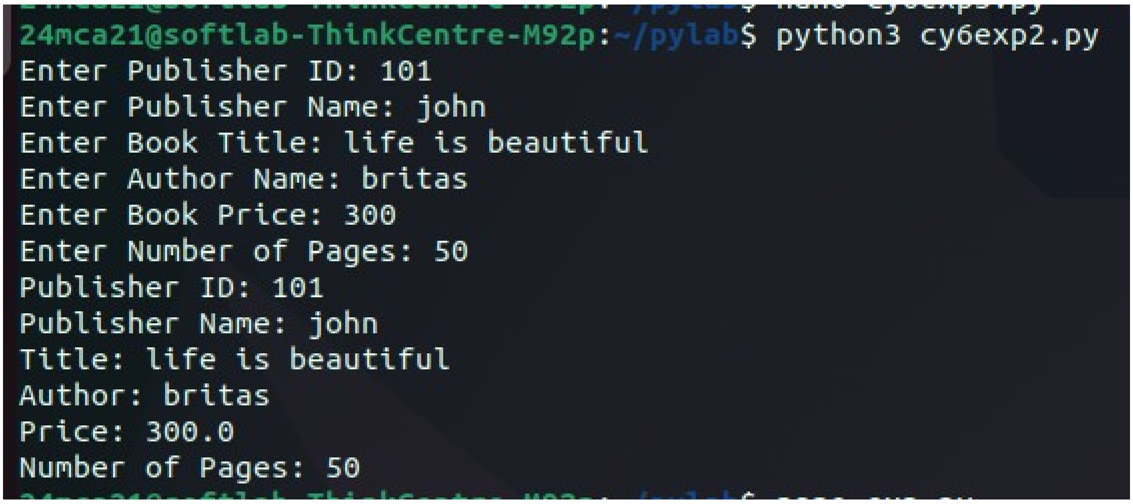
AIM

Create a class Publisher with attributes publisher id and publisher name. Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

SOURCE CODE

```
class Publisher:
    def __init__(self, publisher_id, publisher_name):
        self.publisher_id = publisher_id
        self.publisher_name = publisher_name

    def display_publisher(self):
        print(f"Publisher ID: {self.publisher_id}")
        print(f"Publisher Name: {self.publisher_name}")

class Book(Publisher):
    def __init__(self, publisher_id, publisher_name, title, author):

        super().__init__(publisher_id, publisher_name)
        self.title = title
        self.author = author

    def display_book(self):
        print(f"Title: {self.title}")
        print(f"Author: {self.author}")

class Python(Book):
    def __init__(self, publisher_id, publisher_name, title, author, price, no_of_pages):
        super().__init__(publisher_id, publisher_name, title, author)
        self.price = price
        self.no_of_pages = no_of_pages

    def display_python_book(self):
        print(f"Price: {self.price}")
        print(f"Number of Pages: {self.no_of_pages}")

if __name__ == "__main__":
    publisher_id = input("Enter Publisher ID: ")
    publisher_name = input("Enter Publisher Name: ")
    title = input("Enter Book Title: ")
    author = input("Enter Author Name: ")
    price = float(input("Enter Book Price: "))
    no_of_pages = int(input("Enter Number of Pages: "))

    python_book = Python(publisher_id, publisher_name, title, author, price, no_of_pages)
    python_book.display_publisher()
    python_book.display_book()
    python_book.display_python_book()
```

OUTPUT



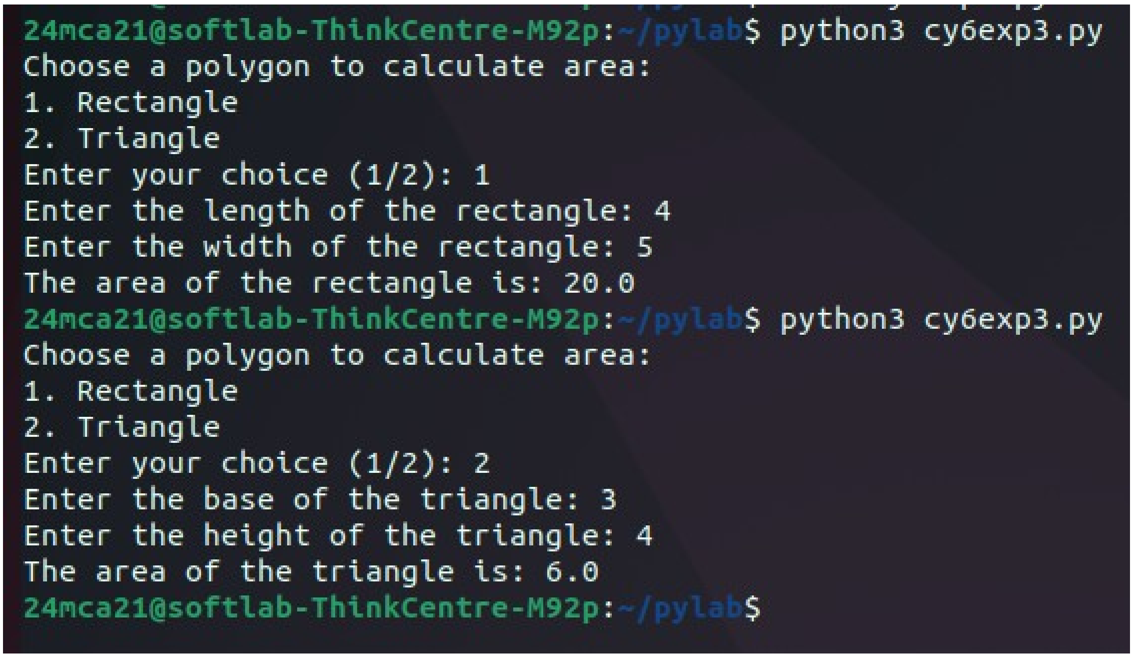
```
24mca21@softlab-ThinkCentre-M92p:~/pylab$ python3 cy6exp2.py
Enter Publisher ID: 101
Enter Publisher Name: john
Enter Book Title: life is beautiful
Enter Author Name: britas
Enter Book Price: 300
Enter Number of Pages: 50
Publisher ID: 101
Publisher Name: john
Title: life is beautiful
Author: britas
Price: 300.0
Number of Pages: 50
```

PROGRAM NO:3
AIM
Write a program that has an abstract class Polygon. Derive two classes Rectangle and Triangle from Polygon and write methods to get the details of their dimensions and hence calculate the area.

SOURCE CODE

```
from abc import ABC, abstractmethod

class Polygon(ABC):
    @abstractmethod
    def get_dimensions(self):
        pass

    @abstractmethod
    def calculate_area(self):
        pass

class Rectangle(Polygon):
    def __init__(self):
        self.length = 0
        self.width = 0

    def get_dimensions(self):
        self.length = float(input("Enter the length of the rectangle: "))
        self.width = float(input("Enter the width of the rectangle: "))

    def calculate_area(self):
        area = self.length * self.width
        return area

class Triangle(Polygon):
    def __init__(self):
        self.base = 0
        self.height = 0

    def get_dimensions(self):
        self.base = float(input("Enter the base of the triangle: "))
        self.height = float(input("Enter the height of the triangle: "))

    def calculate_area(self):
        area = 0.5 * self.base * self.height
        return area

if __name__ == "__main__":
    print("Choose a polygon to calculate area:")
    print("1. Rectangle")
    print("2. Triangle")
    choice = input("Enter your choice (1/2): ")

    if choice == "1":
        rectangle = Rectangle()
        rectangle.get_dimensions()
        area = rectangle.calculate_area()
        print(f"The area of the rectangle is: {area}")

    elif choice == "2":
        triangle = Triangle()
        triangle.get_dimensions()
        area = triangle.calculate_area()
        print(f"The area of the triangle is: {area}")

    else:
        print("Invalid choice. Please select either 1 or 2.")
```

OUTPUT



```
24mca21@softlab-ThinkCentre-M92p:~/pylab$ python3 cy6exp3.py
Choose a polygon to calculate area:
1. Rectangle
2. Triangle
Enter your choice (1/2): 1
Enter the length of the rectangle: 4
Enter the width of the rectangle: 5
The area of the rectangle is: 20.0
24mca21@softlab-ThinkCentre-M92p:~/pylab$ python3 cy6exp3.py
Choose a polygon to calculate area:
1. Rectangle
2. Triangle
Enter your choice (1/2): 2
Enter the base of the triangle: 3
Enter the height of the triangle: 4
The area of the triangle is: 6.0
24mca21@softlab-ThinkCentre-M92p:~/pylab$
```

PROGRAM NO:4

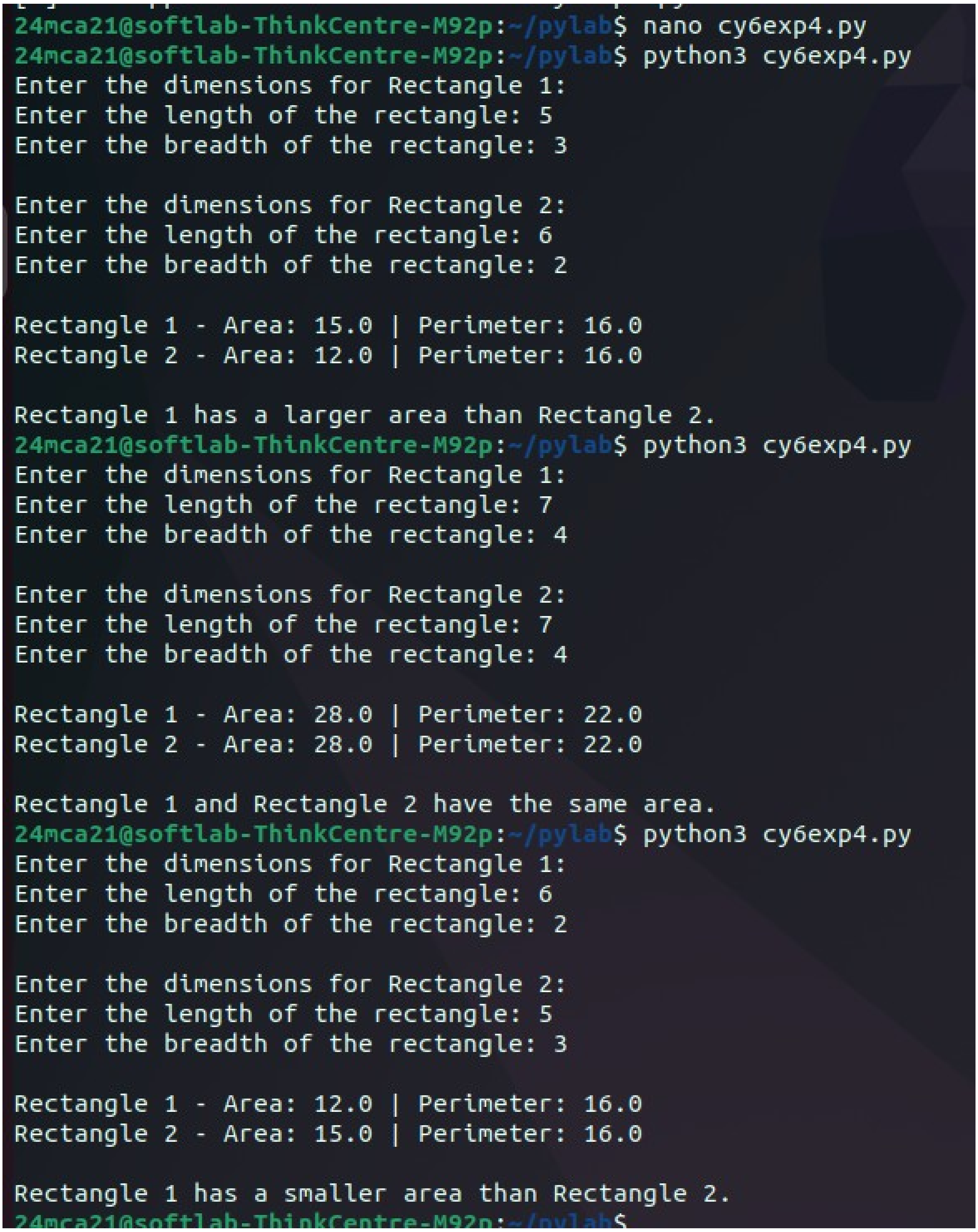
AIM

Create a Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.

SOURCE CODE

```
class Rectangle:
    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth

    def area(self):
        """Method to calculate the area of the rectangle."""
        return self.length * self.breadth

    def perimeter(self):
        """Method to calculate the perimeter of the rectangle."""
        return 2 * (self.length + self.breadth)



if __name__ == "__main__":
    print("Enter the dimensions for Rectangle 1:")
    length1 = float(input("Enter the length of the rectangle: "))
    breadth1 = float(input("Enter the breadth of the rectangle: "))
    rect1 = Rectangle(length1, breadth1)
    print("\nEnter the dimensions for Rectangle 2:")
    length2 = float(input("Enter the length of the rectangle: "))
    breadth2 = float(input("Enter the breadth of the rectangle: "))
    rect2 = Rectangle(length2, breadth2)
    print(f"\nRectangle 1 - Area: {rect1.area()} | Perimeter: {rect1.perimeter()}")
    print(f"Rectangle 2 - Area: {rect2.area()} | Perimeter: {rect2.perimeter()}")
    area1 = rect1.area()
    area2 = rect2.area()

    if area1 < area2:
        print("\nRectangle 1 has a smaller area than Rectangle 2.")
    elif area1 == area2:
        print("\nRectangle 1 and Rectangle 2 have the same area.")
    else:
        print("\nRectangle 1 has a larger area than Rectangle 2.")
```

OUTPUT



```
24mca21@softlab-ThinkCentre-M92p:~/pylab$ nano cy6exp4.py
24mca21@softlab-ThinkCentre-M92p:~/pylab$ python3 cy6exp4.py
Enter the dimensions for Rectangle 1:
Enter the length of the rectangle: 5
Enter the breadth of the rectangle: 3

Enter the dimensions for Rectangle 2:
Enter the length of the rectangle: 6
Enter the breadth of the rectangle: 2

Rectangle 1 - Area: 15.0 | Perimeter: 16.0
Rectangle 2 - Area: 12.0 | Perimeter: 16.0

Rectangle 1 has a larger area than Rectangle 2.
24mca21@softlab-ThinkCentre-M92p:~/pylab$ python3 cy6exp4.py
Enter the dimensions for Rectangle 1:
Enter the length of the rectangle: 7
Enter the breadth of the rectangle: 4

Enter the dimensions for Rectangle 2:
Enter the length of the rectangle: 7
Enter the breadth of the rectangle: 4

Rectangle 1 - Area: 28.0 | Perimeter: 22.0
Rectangle 2 - Area: 28.0 | Perimeter: 22.0

Rectangle 1 and Rectangle 2 have the same area.
24mca21@softlab-ThinkCentre-M92p:~/pylab$ python3 cy6exp4.py
Enter the dimensions for Rectangle 1:
Enter the length of the rectangle: 6
Enter the breadth of the rectangle: 2

Enter the dimensions for Rectangle 2:
Enter the length of the rectangle: 5
Enter the breadth of the rectangle: 3

Rectangle 1 - Area: 12.0 | Perimeter: 16.0
Rectangle 2 - Area: 15.0 | Perimeter: 16.0

Rectangle 1 has a smaller area than Rectangle 2.
24mca21@softlab-ThinkCentre-M92p:~/pylab$
```

PROGRAM NO:5

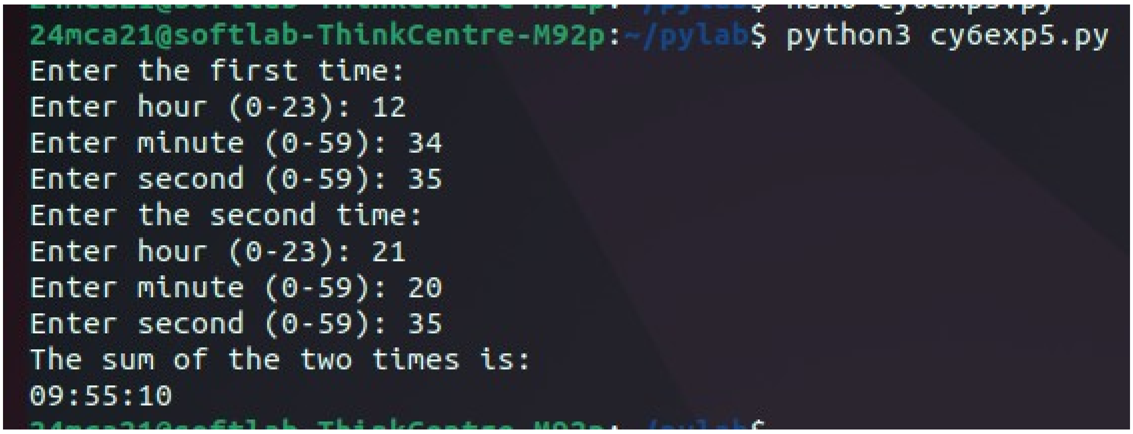
AIM

Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 times.

SOURCE CODE

```
class Time:
    def __init__(self, hour=0, minute=0, second=0):
        self.__hour = hour
        self.__minute = minute
        self.__second = second
    def display_time(self):
        print(f"{self.__hour:02}:{self.__minute:02}:{self.__second:02}")

    def add_times(self, other):
        total_seconds = self.__second + other.__second
        total_minutes = self.__minute + other.__minute + total_seconds // 60
        total_hours = self.__hour + other.__hour + total_minutes // 60
        total_seconds = total_seconds % 60
        total_minutes = total_minutes % 60
        total_hours = total_hours % 24
        return Time(total_hours, total_minutes, total_seconds)

def get_time_input():
    hour = int(input("Enter hour (0-23): "))
    minute = int(input("Enter minute (0-59): "))
    second = int(input("Enter second (0-59): "))
    return Time(hour, minute, second)

print("Enter the first time:")
time1 = get_time_input()

print("Enter the second time:")
time2 = get_time_input()

time3 = time1.add_times(time2)

print("The sum of the two times is:")
time3.display_time()
```

OUTPUT



```
24mca21@softlab-ThinkCentre-M92p:~/pylab$ nano cy6exp5.py
24mca21@softlab-ThinkCentre-M92p:~/pylab$ python3 cy6exp5.py
Enter the first time:
Enter hour (0-23): 12
Enter minute (0-59): 34
Enter second (0-59): 35
Enter the second time:
Enter hour (0-23): 21
Enter minute (0-59): 20
Enter second (0-59): 35
The sum of the two times is:
09:55:10
24mca21@softlab-ThinkCentre-M92p:~/pylab$
```