COURSE OUTCOME 4

DATE : 27-11-2023

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

Program

```
class Rectangle:
  def __init__(self,length,breadth):
    self.length=length
    self.breadth=breadth
  def area(self):
    return self.length * self.breadth
 def perimeter(self):
   return 2 * (self.length + self.breadth)
 def compare_area(self, other_rectangle):
   if self.area() > other_rectangle.area():
     return "The first rectangle has a larger area."
   elif self.area() < other_rectangle.area():
     return "The second rectangle has a larger area."
   else:
     return "Both rectangles have the same area."
print("first rectangle: ")
length=int(input("Enter the length of the rectangle:"))
breadth=int(input("Enter the breadth of the rectangle:"))
```

```
    rectangle1 = Rectangle(length,breadth)
    print("Area of Rectangle 1:", rectangle1.area())
    print("Perimeter of Rectangle 1:", rectangle1.perimeter())
    print("second rectangle: ")
    length=int(input("Enter the length of the rectangle:"))
    breadth=int(input("Enter the breadth of the rectangle:"))
  rectangle2 = Rectangle(length,breadth)
  print("Area of Rectangle 1:", rectangle2.area())
  print("Perimeter of Rectangle 1:", rectangle2.perimeter())
  comparison_result = rectangle1.compare_area(rectangle2)
  print(comparison_result)
```

Output

first rectangle:

Enter the length of the rectangle:2

Enter the breadth of the rectangle:5

Area of Rectangle 1: 10

Perimeter of Rectangle 1: 14

second rectangle:

Enter the length of the rectangle:3

Enter the breadth of the rectangle:6

Area of Rectangle 1: 18

Perimeter of Rectangle 1: 18

The second rectangle has a larger area.

DATE : 27-11-2023

2. **Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.**

Program

```
class BankAccount:
 def __init__(self, account_number, account_holder_name, account_type, balance=0):
   self.account_number = account_number
   self.account_holder_name = account_holder_name
   self.account_type = account_type
   self.balance = balance
 def deposit(self, amount):
    if amount > 0:
      self.balance = self.balance + amount
      print("\nDeposition Successful!")
   else:
      print("\nInvalid amount!")
 def withdraw(self, amount):
   if 0 < amount < self.balance:
     self.balance = self.balance - amount
     print("Withdrawal successful")
     print("New Balance : ",self.balance)
```

```
        elif amount > self.balance:
            print("Not possible to withdraw. Insufficient funds.")
        else:
            print("Invalid amount!")
    def get_balance(self):
        return self.balance
account1 = BankAccount("123456", "Abhimanue", "Savings", 10000)
print("current balance:",account1.get_balance())
deposit_amount = float(input("Enter the deposit amount: "))
account1.deposit(deposit_amount)
withdrawal_amount = float(input("Enter the withdrawal amount: "))
account1.withdraw(withdrawal_amount)
account1.get_balance()
```

Output

```
current balance: 10000
Enter the deposit amount: 10000
Deposition Successful!
Enter the withdrawal amount: 5000
Withdrawal successful
New Balance : 15000
```

DATE : 27-11-2023

3. **Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.**

Program

```
class Rectangle:
  def __init__(self, length, width):
    self._length = length
    self._width = width
  def area(self):
    return self._length * self._width
  def __lt__(self, other):
    return self.area() < other.area()
print("First rectangle: ")
l=int(input("Enter the length of the rectangle:"))
b=int(input("Enter the breadth of the rectangle:"))
rectangle1 = Rectangle(l, b)
print("Second rectangle: ")
l=int(input("Enter the length of the rectangle:"))
b=int(input("Enter the breadth of the rectangle:"))
```

```
rectangle2 = Rectangle(l, b)

if rectangle1 < rectangle2:

print("Area of Rectangle 1 is smaller than the area of Rectangle 2.")

elif rectangle1 > rectangle2:

print("Area of Rectangle 1 is larger than the area of Rectangle 2.")

else:

print("Both rectangles have the same area.")
```

Output

```
First rectangle:

Enter the length of the rectangle:5

Enter the breadth of the rectangle:2

Second rectangle:

Enter the length of the rectangle:3

Enter the breadth of the rectangle:6

Area of Rectangle 2 is larger than the area of Rectangle 1.
```

DATE : 29-11-2023

4. **Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.**

Program

```
class Time:
  def __init__(self, hour=0, minute=0, second=0):
    self._hour = hour
    self._minute = minute
    self._second = second
  def __add__(self, other):
    total_seconds = self._hour * 3600 + self._minute * 60 + self._second + \
    other._hour * 3600 + other._minute * 60 + other._second
    new_hour, remainder = divmod(total_seconds, 3600)
    new_minute, new_second = divmod(remainder, 60)
    return Time(new_hour, new_minute, new_second)
  def __str__(self):
    return f"{self._hour:02d}:{self._minute:02d}:{self._second:02d}"
a=int(input("Enter the hour of time1:"))
b=int(input("Enter the minute of time1:"))
c=int(input("Enter the second of time1:"))
```

```
x=int(input("Enter the hour of time2:"))

y=int(input("Enter the minute of time2:"))

z=int(input("Enter the second of time2:"))

time1 = Time(a,b,c)

time2 = Time(x,y,z)

sum_time = time1 + time2

print("Time 1:", time1)

print("Time 2:", time2)

print("Sum of Time 1 and Time 2:", sum_time)
```

Output

```
Enter the hour of time1:2

Enter the minute of time1:30

Enter the second of time1:10

Enter the hour of time2:4

Enter the minute of time2:00

Enter the second of time2:30

Time 1: 02:30:10

Time 2: 04:00:30

Sum of Time 1 and Time 2: 06:30:40
```

DATE : 29-11-2023

5. **Create a class 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.**

Program

```
class Publisher:
  def __init__(self, name):
    self.name = name
class Book(Publisher):
  def __init__(self, name, title, author):
    super().__init__(name)
    self.title = title
    self.author = author
  def display_info(self):
    print("Publisher:", self.name)
    print("Title:", self.title)
    print("Author:", self.author)
class Python(Book):
  def __init__(self, name, title, author, price, no_of_pages):
    super().__init__(name, title, author)
    self.price = price
    self.no_of_pages = no_of_pages
```

```
  def display_info(self):
  super().display_info()
  print("Price:", self.price)
  print("Number of Pages:", self.no_of_pages)
publisher=input("Enter the name of publisher:")
book_name=input("Enter the name of book:")
author=input("Enter the name of author:")
book_price=input("Enter the price of book:")
book_pages=input("Enter the number of pages of the book:")
python_book = Python(publisher, book_name, author, book_price, book_pages)
python_book.display_info()
```

Output

```
Enter the name of publisher:Universities Press
Enter the name of book:Wings of fire
Enter the name of author:A.P.J.Abdul Kalam
Enter the price of book:385
Enter the number of pages of the book:190
Publisher: Universities Press
Title: Wings of fire
Author: A.P.J.Abdul Kalam
Price: 385
Number of Pages: 190
```