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 length of rectangle"))
breadth=int(input("Enter breadth of rectangle"))
rectangle1=Rectangle(length,breadth)
print("Area of rectangle1:",rectangle1.area())
print("Perimeter of rectangle1",rectangle1.perimeter())

print("Second rectangle")
length2=int(input("Enter length of rectangle"))
breadth2=int(input("Enter breadth of rectangle"))
rectangle2=Rectangle(length2,breadth2)
print("Area of rectangle2:",rectangle2.area())
print("Perimeter of rectangle2:",rectangle2.perimeter())
comparison_result = rectangle1.compare_area(rectangle2)
```

```
print(comparison_result)
```

Output

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$ python3 rect.py
First Rectangle
Enter length of rectangle2
Enter breadth of rectangle3
Area of rectangle1: 6
Perimeter of rectangle1 10
Second rectangle
Enter length of rectangle4
Enter breadth of rectangle5
Area of rectangle2: 20
Perimeter of rectangle2: 18
The second rectangle has a larger area
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$
```

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,accno,name,acctype,balance=0):
   self.accno=accno
   self.name=name
   self.acctype=acctype
   self.balance=balance

  def deposit(self,amount):
   if(amount>0):
    self.balance=self.balance+amount
    print("Successfull deposit of",amount)
    print("New balance:",self.balance)
   else:
    print("Not successfull")

  def withdraw(self,amount):
   if(0<amount<self.balance):
      self.balance=self.balance-amount
   elif(amount>self.balance):
      print("Not possible to withdraw")
   else:
      print("Invalid")

  def getbalance(self):
   print("Current balance:",self.balance)

accno=int(input("Enter the account number:"))
name=input("Enter your name:")
acctype=input("Enter the type of account:")
account1=Bankaccount(accno,name,acctype)
```

```
print("MENU")
while True:
print("1.Deposit \n 2.Withdraw \n 3.Balance \n 4.Exit")
 ch=int(input("Enter your choice:"))
 if(ch==1):
    damount=int(input("Enter the amount to be deposited:"))
    account1.deposit(damount)
 elif(ch==2):
     wamount=int(input("Enter the amount to be withdraw:"))
     account1.withdraw(wamount)
 elif(ch==3):
    account1.getbalance()
 elif(ch==4):
   exit(0)
 else:
   print("Wrong choice")
```

Output

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$ python3 bnk.py
Enter the account number:123
Enter your name:Anu
Enter the type of account:savings
MENU
1.Deposit
2.Withdraw
3.Balance
4.Exit
Enter your choice:1
Enter the amount to be deposited:25000
Successfull deposit of 25000
New balance: 25000
1.Deposit
2.Withdraw
3.Balance
4.Exit
Enter your choice:3
Current balance: 25000
1.Deposit
2.Withdraw
3.Balance
4.Exit
Enter your choice:2
Enter the amount to be withdraw:1000
1.Deposit
2.Withdraw
3.Balance
4.Exit
Enter your choice:3
Current balance: 24000
1.Deposit
2.Withdraw
3.Balance
4.Exit
Enter your choice:4
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$
```

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 ")
length=int(input("Enter length of rectangle"))
width=int(input("Enter breadth of rectangle"))
rectangle1=Rectangle(length,width)
print("Area of rectangle1:",rectangle1.area())
print("Second rectangle")
length2=int(input("Enter length of rectangle"))
width2=int(input("Enter breadth of rectangle"))
rectangle2=Rectangle(length2,width2)
print("Area of rectangle2:",rectangle2.area())
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

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$ python3 pvt.py
First Rectangle
Enter length of rectangle5
Enter breadth of rectangle7
Area of rectangle1: 35
Second rectangle
Enter length of rectangle4
Enter breadth of rectangle8
Area of rectangle2: 32
Area of Rectangle 1 is larger than the area of Rectangle 2.
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$
```

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}"

h1=int(input("Enter the hour"))
m1=int(input("Enter the minute"))
s1=int(input("Enter the second"))
time1=Time(h1,m1,s1)
h2=int(input("Enter the hour"))
m2=int(input("Enter the minute"))
s2=int(input("Enter the second"))
time2=Time(h2,m2,s2)
sum_time = time1 + time2
print("Time 1:", time1)
print("Time 2:", time2)
print("Sum of Time 1 and Time 2:", sum_time)
```

Output

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$ python3
time.py
Enter the hour5
Enter the minute5
Enter the second5
Enter the hour7
Enter the minute30
Enter the second15
Time 1: 05:05:05
Time 2: 07:30:15
Sum of Time 1 and Time 2: 12:35:20
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$
```

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): # Method overriding
  super().display_info() # Invoking the base class (Book) method
  print("Price:", self.price)
  print("Number of Pages:", self.no_of_pages)

name_p=input("Enter the name of publisher:")
title_b=input("Enter the title of book:")
name_a=input("Enter the name of author:")
price=int(input("Enter the price of book:"))
noofpages=int(input("Enter the no of pages of book:"))
python_book = Python(name_p,title_b,name_a,price,noofpages )
python_book.display_info()
```

Output

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$ python3
book.py
Enter the name of publisher:Bloomsbury publishing
Enter the title of book:Harry Potter
Enter the name of author:J K Rowling
Enter the price of book:1199
Enter the no of pages of book:766
Publisher: Bloomsbury publishing
Title: Harry Potter
Author: J K Rowling
Price: 1199
Number of Pages: 766
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/python$
```