M.No	Date	IDX	PROGRAM	Sign & Marks
11		Python Programs on OOPs		
		11.1	Class and object creation --- person class	
		11.2	Bank Transactions	
		11.3	Shopping Cart	
		11.4	Simple Calculator	
		11.5	Shapes class with area and perimeter	
		11.6	Polymorphism and inheritance ,overriding	

AIM : Write a Python program to create a person class. Include attributes like name, country and date of birth. Implement a method to determine the person's age.

PROGRAM 1

```
import datetime
from datetime import date
class Person:
   def __init__(self, name, state, date_of_birth):
      self.name=name
      self.state=state
      self.date_of_birth=date_of_birth
   def calculate_age(self):
      today=date.today()
      age=today.year - self.date_of_birth.year
      if today< date(today.year, self.date_of_birth.month,self.date_of_birth.day):
         age=age-1
      return age

n=input("Enter your Name : ")
c=input("Enter your state : ")
d=input(" enter your date of Birth(date-monthno-completeyear) format : ")
f = '%d-%m-%Y'
dob = datetime.datetime.strptime(d, f)
print("Person Details are : ")
p1=Person(n,c,dob)
print(" Name Of the Person : ",p1.name)
print(" State Of the Person : ",p1.state)
print(" Date of Birth Of the Person : ",p1.date_of_birth.strftime("%d-%m-%Y"))
print(" Age Of the Person : ",p1.calculate_age())
```

OUTPUT

```
Enter your Name : Hema Latha Reddy
Enter your state : Andhra Pradesh
 enter your date of Birth(date-monthno-completeyear) format : 22-03-1975
Person Details are :
 Name Of the Person :  Hema Latha Reddy
 State Of the Person :  Andhra Pradesh
 Date of Birth Of the Person :  22-03-1975
 Age Of the Person :  49
```

AIM : Write a Python program to create a class representing a bank. Include methods for managing customer accounts and transactions.

PROGRAM 2 :

```python
class Bank_Account:
    def __init__(self):
        self.customers = {}
    def create_account(self,acno,bal=0):
        if acno in self.customers:
            print("Account number already Exist ")
        else:
            self.customers[acno]=bal
            print(" Account created successfully. ")

    def deposit(self,acno,amt):
        if acno in self.customers:
            self.customers[acno]+=amt
            print(" Deposit successful. ")
        else:
            print(" Account Number does not exist. ")

    def withdraw(self,acno,amt):
        if acno in self.customers:
            if self.customers[acno]>=amt:
                self.customers[acno] -= amt
                print(" Withdrawal successful. ")
            else:
                print(" Withdrawal Fail due to Insufficient funds. ")
        else:
            print(" Account Number does not exist. ")

    def Balance(self,acno):
        if acno in self.customers:
            bal=self.customers[acno]
            print(" Account Balance : ",bal)
        else:
            print(" Account Number does not exist. ")
```

```
s = Bank_Account()
acno="SB-123"
damt1=1000
print(" New a/c No : ",acno,"Deposit Amount : ",damt1)
s.create_account(acno,damt1)
damt2=600
print(" \nDeposit Rs. ",damt2," to a/c No. ",acno)
s.deposit(acno,damt2);
s.Balance(acno)
wamt1=500
print(" \nWithdraw Rs. ",wamt1," from a/c No. ",acno)
s.withdraw(acno,wamt1);
s.Balance(acno)
wamt2=2000
print(" \nWithdraw Rs. ",wamt2," from a/c No. ",acno)
s.withdraw(acno,wamt2);
s.Balance(acno)
```

OUTPUT

```
 New a/c No :  SB-123 Deposit Amount :  1000
 Account created successfully.

Deposit Rs.  600  to a/c No.  SB-123
 Deposit successful.
 Account Balance :  1600

Withdraw Rs.  500  from a/c No.  SB-123
 Withdrawal successful.
 Account Balance :  1100

Withdraw Rs.  2000  from a/c No.  SB-123
 Withdrawal Fail due to Insufficient funds.
 Account Balance :  1100
```

AIM : Write a Python program to create a class representing a shopping cart. Include methods for adding and removing items, and calculating the total price.

PROGRAM 3:

```
class ShoppingCart:
    def __init__(self):
        self.items = []

    def add_item(self,item_name,qty):
        item=(item_name,qty)
        self.items.append(item)

    def remove_item(self,item_name):
        for item in self.items:
            if item[0] == item_name:
                self.items.remove(item)
                break

    def display(self):
        for item in self.items:
            print(item[0],"-",item[1])

    def calculate_total(self):
        total=0
        for item in self.items:
            total +=item[1]
        return total
```

```
cart=ShoppingCart()
cart.add_item("Papaya",10)
cart.add_item("Apple",20)
cart.add_item("Orange",50)
print(" Current Items in Cart are ")
cart.display()
total_qty=cart.calculate_total()
print(" Total Quantity : ",total_qty)
cart.remove_item("Orange")
print("\nUpdated Items in cart after removing Orange is : ")
cart.display()
total_qty=cart.calculate_total()
print(" Total Quantity : ",total_qty)
```

OUTPUT :

```
 Current Items in Cart are
Papaya - 10
Apple - 20
Orange - 50
 Total Quantity :  80

Updated Items in cart after removing Orange is :
Papaya - 10
Apple - 20
 Total Quantity :  30
```

AIM : Write a Python program to create a calculator class. Include methods for basic arithmetic operations.

PROGRAM 4 :

```
class Calculator:
    def add(self, x, y):
        return x+y

    def subtract(self,x,y):
        return x-y
    def multiply(self,x,y):
        return x*y
    def divide(self,x,y):
        if y!= 0:
            return x/y
        else:
            return (" Cannot divide by Zero. ")

calc=Calculator()
a=int(input("Enter the first number : "))
b=int(input("Enter the second number : "))
print("Sum = ",calc.add(a,b))
print("Difference = ",calc.subtract(a,b))
print("Product  = ",calc.multiply(a,b))
print("Division = ",calc.divide(a,b))
```

OUTPUT :

```
Enter the first number : 10
Enter the second number : 5
Sum =  15
Difference =  5
Product  =  50
Division =  2.0
```

AIM : Write a python program to create a class that represents a shape. Include methods to calculate its area and perimeter. Implement subclasses for different shapes like circle, triangle and Rectangle etc.

PROGRAM 5:

```
class Shape:
    def area(self):
        pass

    def perimeter(self):
        pass

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

    def area(self):
        return 22/7 * self.radius**2

    def perimeter(self):
        return 2 * 22/7 * self.radius

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

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

    def perimeter(self):
        return 2 * (self.length + self.width)

class Triangle(Shape):
    def __init__(self, base, height, side1, side2, side3):
        self.base = base
        self.height = height
        self.side1 = side1
        self.side2 = side2
        self.side3 = side3

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

    def perimeter(self):
        return self.side1 + self.side2 + self.side3
```

```
r = 7
circle = Circle(r)
circle_area = circle.area()
circle_perimeter = circle.perimeter()

print("Radius of the circle:", r)
print("Circle Area:", circle_area)
print("Circle Perimeter:", circle_perimeter)
l=5
w=7
rectangle = Rectangle(l, w)
rectangle_area = rectangle.area()
rectangle_perimeter = rectangle.perimeter()

print("\nRectangle: Length =", l, " Width =", w)
print("Rectangle Area:", rectangle_area)
print("Rectangle Perimeter:", rectangle_perimeter)

base = 5
height = 4
s1 = 4
s2 = 3
s3 = 5

print("\nTriangle: Base =", base, " Height =", height, " side1 =", s1, " side2 =", s2, " side3 =", s3)
triangle = Triangle(base, height, s1, s2, s3)
triangle_area = triangle.area()
triangle_perimeter = triangle.perimeter()
print("Triangle Area:", triangle_area)
print("Triangle Perimeter:", triangle_perimeter)
```

OUTPUT:

```
Radius of the circle: 7
Circle Area: 154.0
Circle Perimeter: 44.0

Rectangle: Length = 5  Width = 7
Rectangle Area: 35
Rectangle Perimeter: 24

Triangle: Base = 5  Height = 4  side1 = 4  side2 = 3  side3 = 5
Triangle Area: 10.0
Triangle Perimeter: 12
```

AIM : Write a python program to demonstrate Polymorphism and Inheritance (Method Overriding)

PROGRAM 6:

```
class Vehicle:
   def __init__(self, brand, model, price):
      self.brand = brand
      self.model = model
      self.price = price

   def show(self):
      print('\nDetails:', self.brand, self.model, 'Price:', self.price)

   def max_speed(self):
      print('Vehicle max speed is 160')

   def gear_system(self):
      print('Vehicle has 6 shifter gearbox')

class Car(Vehicle):
   def max_speed(self):
      print('Car max speed is 260')

   def gear_system(self):
      print('Car has Automatic Transmission')

car = Car('Audi', 'R8', 9000000)
car.show()

car.max_speed()
car.gear_system()

vehicle = Vehicle('Nissan', 'Magnite', 550000)
vehicle.show()

vehicle.max_speed()
vehicle.gear_system()
```

OUTPUT: