

# Practical No 1

**Aim : Write A Program In Order To Explain The Concepts Of Class And Objects.**

## 1. Making A Class For Circle

```
#-----Class-----#
# Creating Class Of Circle
class Circle:
    radius=0
    circumference=0
    area=0

    #Circumference Of Circle
    def circumferenceOfCircle(this):
        this.circumference = 2 * 3.14 * this.radius
        return 2 * 3.14 * this.radius

    #Area Of Circle
    def areaOfCircle(this):
        this.area = 3.14 * (this.radius ** 2)
        return 3.14 * (this.radius ** 2)

    #Display Curcumference
    def displayCircumference(this):
        print(this.circumference)

    #Display Area
    def displayArea(this):
        print(this.area)

#-----Code-----#

# Making Object
abc = Circle()

# User choice
choice = "y"

while choice == "y":
    # Giving Radius To Object
    abc.radius = int(input("Enter Radius : "))
```

#Calling Function To Calculate Area And Circumference

```
abc.circumferenceOfCircle()
abc.areaOfCircle()
# Printing Circumference , Area
print ("Circumference Of Circle")
abc.displayCircumference()
print("Area Of Circle")
abc.displayArea()
```

```
# Asking User If Wants To Continue
choice = input("Want To Continue : ")
```

```
Enter Radius : 7
Circumference Of Circle
43.96
Area Of Circle
153.86
Want To Continue : y
Enter Radius : 9
Circumference Of Circle
56.52
Area Of Circle
254.34
Want To Continue : n
```

## 2. Making Bike Class

```
#-----Class-----#
```

```
class Bike:
    name = ""
    gear = 0
```

```
#-----Code-----#
```

```
B1 = Bike()
B1.name = "Ducati"
B1.gear = 5
```

```
B2 = Bike()
B2.name = "kawasaki"
B2.gear = 7
```

```
B3 = Bike()
B3.name = "BMW"
B3.gear = 6
```

```
B4 = Bike()
B4.name = "Nissan"
B4.gear = 5
```

```

print("-----Bike-----")
print("Your bike name",B1.name,"has gear ",B1.gear)
print("-----")
print("Your bike name",B2.name,"has gear ",B2.gear)
print("-----")
print("Your bike name",B3.name,"has gear ",B3.gear)
print("-----")
print("Your bike name",B4.name,"has gear ",B4.gear)
print("-----")

```

```

-----Bike-----
Your bike name Ducati has gear 5
-----
Your bike name kawasaki has gear 7
-----
Your bike name BMW has gear 6
-----
Your bike name Nissan has gear 5
-----

```

### 3. Constructor And Destructor

#### a. Default Constructor

# Creating Default Constructor

```

class Car :
    def __init__(self):
        self.name = "BMW"
        self.model = "X1"
        self.year = 2020

```

```

car = Car()

```

```

print("-----")
print("Default Constructor")
print("-----")
print(car.name)
print(car.model)
print(car.year)
# Updating Value
car.name = "Audi"
print(car.name)

```

```

-----
Default Constructor
-----
BMW
X1
2020
Audi

```

## b. Parameterized Constructor

```
print("-----")
print("Parameterized Constructor")
print("-----")
# Creating Parameterized Constructor
class Car :
    def __init__(self,name,model,year):
        self.name = name
        self.model = model
        self.year = year

car = Car("BMW","X1",2020)

print("-----Normal Display-----")
print(car.name)
print(car.model)
print(car.year)
```

```
-----
Parameterized Constructor
-----
```

```
-----Normal Display-----
BMW
X1
2020
```

## c. Display Function

```
print("-----")
print("Parameterized Constructor")
print("-----")
# Creating Parameterized Constructor
class Car :
    def __init__(self,name,model,year):
        self.name = name
        self.model = model
        self.year = year
    def display(self):
        print("Car Name : %s\nCar Model : %s\nCar Year : %d"%(self.name,self.model,self.year))

car = Car("BMW","X1",2020)

print("-----Normal Display-----")
print(car.name)
print(car.model)
print(car.year)
print("-----Through Display Function-----")

car.display()
```

```
-----  
Parameterized Constructor  
-----
```

```
-----Normal Display-----
```

```
BMW
```

```
X1
```

```
2020
```

```
-----Through Display Function-----
```

```
Car Name : BMW
```

```
Car Model : X1
```

```
Car Year : 2020
```

#### **d. Destructor**

```
# Creating Destructor
```

```
print("-----")
```

```
print("Destructor")
```

```
print("-----")
```

```
class Car :
```

```
    def __init__(self,name,model,year):
```

```
        self.name = name
```

```
        self.model = model
```

```
        self.year = year
```

```
    def __del__(self): # Destructor
```

```
        print("Object Destroyed")
```

```
car = Car("BMW","X1",2020)
```

```
print(car.name)
```

```
print(car.model)
```

```
print(car.year)
```

```
del car
```

```
-----  
Destructor  
-----
```

```
BMW
```

```
X1
```

```
2020
```

```
Object Destroyed
```

## 4. Project Of Class

```
# Creating Employee Class
class Employee:
    def __init__(self,id,name):
        self.id = id
        self.name = name

    def display(self):
        print("ID : %d\nName : %s"%(self.id,self.name))

    def __del__(self):
        print("Object Destroyed")

# Taking Input For Number Of Employees
numOfEmployees = 0

# Handling Invalid Input
while numOfEmployees < 1:
    numOfEmployees = int(input("Enter Number Of Employees : "))

# Creating List Of Employees
employees = []

# Taking Input For Each Employee
for i in range(numOfEmployees):
    print("Employee %d"%(i+1))
    id = int(input("Enter ID : "))
    name = input("Enter Name : ")
    employees.append(Employee(id,name))

# Displaying Each Employee
for employee in employees:
    employee.display()
```

```
Enter Number Of Employees : 2
Employee 1
Enter ID : 1
Enter Name : John Doe
Employee 2
Enter ID : 2
Enter Name : Jane Doe
ID : 1
Name : John Doe
ID : 2
Name : Jane Doe
Object Destroyed
Object Destroyed
```