BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai – 400058-India
Department of Computer Engineering

Name	Sujal Dingankar
UID no.	2024301005
Experiment No.	2

AIM:	**OOPs in Python**
Program 1	
PROBLEM STATEMENT :	Hospital Management System: Problem Statement: Design a healthcare system with a base class Person that includes name and age. Derive classes Patient and Doctor from Person. Patient should have an attribute medical_history, while Doctor should have an attribute specialty. Simulate a hospital management system.

PROGRAM:	class Person:
	  def __init__(self, name, age):
	  self.name = name
	  self.age = age
	
	class Doctor(Person):
	  def __init__(self, name, age, speciality, fees):
	  super().__init__(name, age)
	  self.speciality = speciality
	  self.fees = fees
	
	class Patient(Person):
	  def __init__(self, name, age, medical_history):
	  super().__init__(name, age)
	  self.medical_history = medical_history
	
	print("Welcome to the State Hospital!")
	
	d1 = Doctor("Avinash", 30, "Orthopedia", 80000)
	d2 = Doctor("Kadam", 62, "Cardio", 67500)
	d3 = Doctor("Deshmukh", 45, "Ophthalmologist", 20000)

```
d4 = Doctor("Rao", 26, "General", 55000)

Doctors = [d1, d2, d3, d4]

str_input = 'yes'
while str_input.lower() == 'yes':
    name = input("Enter your name: ")
    age = int(input("Enter your age: "))
    var = int(input("Enter the following:\n1. Add a new Doctor\n2. Get
Medical treatment\n"))

    if var == 1:
        # Improving the prompt when adding a new doctor
        doctor_name = input("Enter the doctor's name: ")
        doctor_age = int(input("Enter the doctor's age: "))
        speciality = input("Enter the doctor's speciality: ")
        fees = int(input("Enter the doctor's fees: "))
        print(doctor_name, "is added to the Hospital as a doctor.\n")
        new_doctor = Doctor(doctor_name, doctor_age, speciality, fees)
        Doctors.append(new_doctor)

    elif var == 2:
        med = input("Enter your medical history: ")
        p1 = Patient(name, age, med)

        if med in ["cough", "sneeze", "fever"]:
            print("\nYou will get treatment from")
            print("Name:", d4.name, "\nAge:", d4.age, "\nSpeciality:",
d4.speciality, "\nFees:", d4.fees)

        elif med in ["heart pain", "chest pain", "heart attack"]:
            print("\nYou will get treatment from")
            print("Name:", d2.name, "\nAge:", d2.age,
"\nSpeciality:", d2.speciality, "\nFees:", d2.fees)

        elif med in ["bone disorder", "broken bone",
"ligaments"]:
```

```
        print("\nYou will get treatment from")
        print("Name:", d1.name, "\nAge:", d1.age, "\nSpeciality:",
d1.speciality, "\nFees:", d1.fees)

    elif med in ["eye problem", "myopia", "farsightedness"]:
        print("\nYou will get treatment from")
        print("Name:", d3.name, "\nAge:", d3.age, "\nSpeciality:",
d3.speciality, "\nFees:", d3.fees)

    else:
        print("Sorry, we currently don't have any doctor related to your
medical history\n")

  str_input = input("Do you want to continue (yes/no)? ")
```

RESULT:

```
Welcome to the State Hospital!
Enter your name: Sujal Dingankar
Enter your age: 20
Enter the following:
1. Add a new Doctor
2. Get Medical treatment
1
```

```
Enter the doctor's name: Ganesh Seth
Enter the doctor's age: 19
Enter the doctor's speciality: Cardio
Enter the doctor's fees: 2000
Ganesh Seth is added to the Hospital as a doctor.

Do you want to continue (yes/no)? yes
Enter your name: Harsha
Enter your age: 19
Enter the following:
1. Add a new Doctor
2. Get Medical treatment
2
Enter your medical history: fever
```

```
You will get treatment from
Name: Rao
Age: 26
Speciality: General
Fees: 55000
Do you want to continue (yes/no)? yes
Enter your name: Shreeya
Enter your age: 21
Enter the following:
1. Add a new Doctor
2. Get Medical treatment
2
Enter your medical history: bone disorder

You will get treatment from
Name: Avinash
Age: 30
Speciality: Orthopedia
Fees: 80000
Do you want to continue (yes/no)? no


...Program finished with exit code 0
Press ENTER to exit console.
```

Program 2	
PROBLEM STATEMENT :	Create a Product class with private attributes for product_name and quantity_in_stock.Provide methods to adjust stock levels and retrieve product information.

PROGRAM:	class Product:
	  def __init__(self, product_name, quantity_in_stock, product_id):
	  self.product_name = product_name
	  self.quantity_in_stock = quantity_in_stock
	  self.product_id = product_id
	
	  def reduceStock(self, value):
	  if value > self.quantity_in_stock:
	  print("Not enough stock\n")

```
        else:
            self.quantity_in_stock -= value
            print("Remaining stock is:", self.quantity_in_stock)

    def increaseStock(self, value):
        self.quantity_in_stock += value
        print("No of stocks is:", self.quantity_in_stock)

    def productInformation(self):
        print("Product name:", self.product_name)
        print("Product stock in warehouse:", self.quantity_in_stock)


# Initialize the Products list
Products = []

str_input = 'yes'

while str_input.lower() == 'yes':
    n = int(input("Choose respective option:\n1) Add Product\n2) Update
Stock quantity\n3) Get Product Information\n"))

    match n:
        case 1:
            name = input("Enter name of product: ")
            quantity = int(input("Enter quantity: "))
            prod_id = int(input("Enter product id: "))
            Products.append(Product(name, quantity, prod_id))

        case 2:
            name = input("Enter product name you want to update: ")
            m = int(input("Choose: \n1) Add Stock\n2) Remove
products\n"))
            value = int(input("Enter no of stocks: "))

            if m == 1:
                for product in Products:
                    if product.product_name == name:
```

```
                product.increaseStock(value)
                break
        else:
            for product in Products:
                if product.product_name == name:
                    product.reduceStock(value)
                    break

    case 3:
        name = input("Enter name of product: ")
        for product in Products:
            if product.product_name == name:
                product.productInformation()
                break

str_input = input("Do you wish to continue (yes/no)\n")
```

RESULT:

```
Choose respective option:
1) Add Product
2) Update Stock quantity
3) Get Product Information
1
```

```
Enter name of product: Detol
Enter quantity: 25
Enter product id: 202001
Do you wish to continue (yes/no)
yes
Choose respective option:
1) Add Product
2) Update Stock quantity
3) Get Product Information
1
Enter name of product: ToothBrush
Enter quantity: 5
Enter product id: 202002
Do you wish to continue (yes/no)
yes
Choose respective option:
1) Add Product
2) Update Stock quantity
3) Get Product Information
2
Enter product name you want to update: Detol
Choose:
1) Add Stock
2) Remove products
1
Enter no of stocks: 10
No of stocks is: 35
Do you wish to continue (yes/no)
yes
Choose respective option:
1) Add Product
2) Update Stock quantity
3) Get Product Information
2
```

```
Enter product name you want to update: ToothBrush
Choose:
1) Add Stock
2) Remove products
2
Enter no of stocks: 2
Remaining stock is: 3
Do you wish to continue (yes/no)
```

```
yes
Choose respective option:
1) Add Product
2) Update Stock quantity
3) Get Product Information
3
Enter name of product: Detol
Product name: Detol
Product stock in warehouse: 35
Do you wish to continue (yes/no)
yes
Choose respective option:
1) Add Product
2) Update Stock quantity
3) Get Product Information
3
Enter name of product: ToothBrush
Product name: ToothBrush
Product stock in warehouse: 3
Do you wish to continue (yes/no)
no


...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSION:	In this experiment, I studied the important OOPs concepts in python and successfullyimplemented them in solving various problem statements.