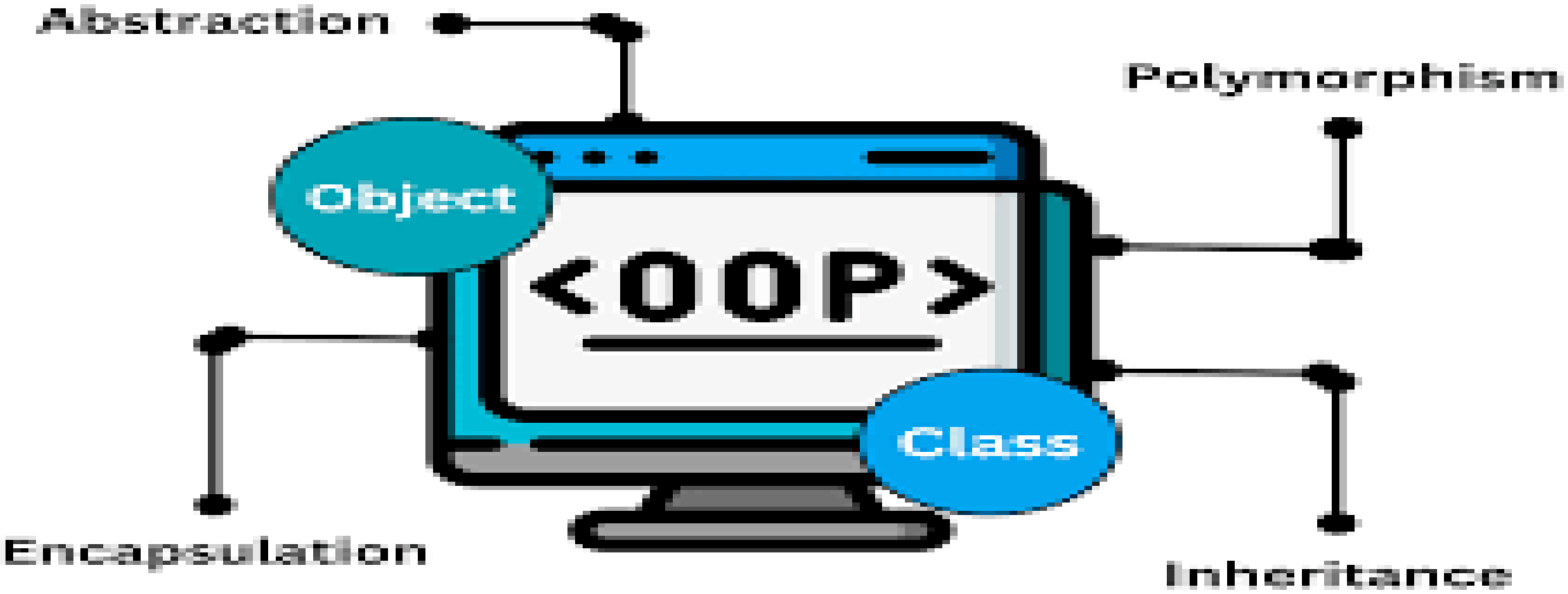
Object Oriented Programming with Java (OOPJ)

Session 3: Operators & Basics

Kiran Waghmare


Abstraction
Polymorphism
Object
<OOP>
Class
Encapsulation
Inheritance

class

car

methods

refuel() getFuel

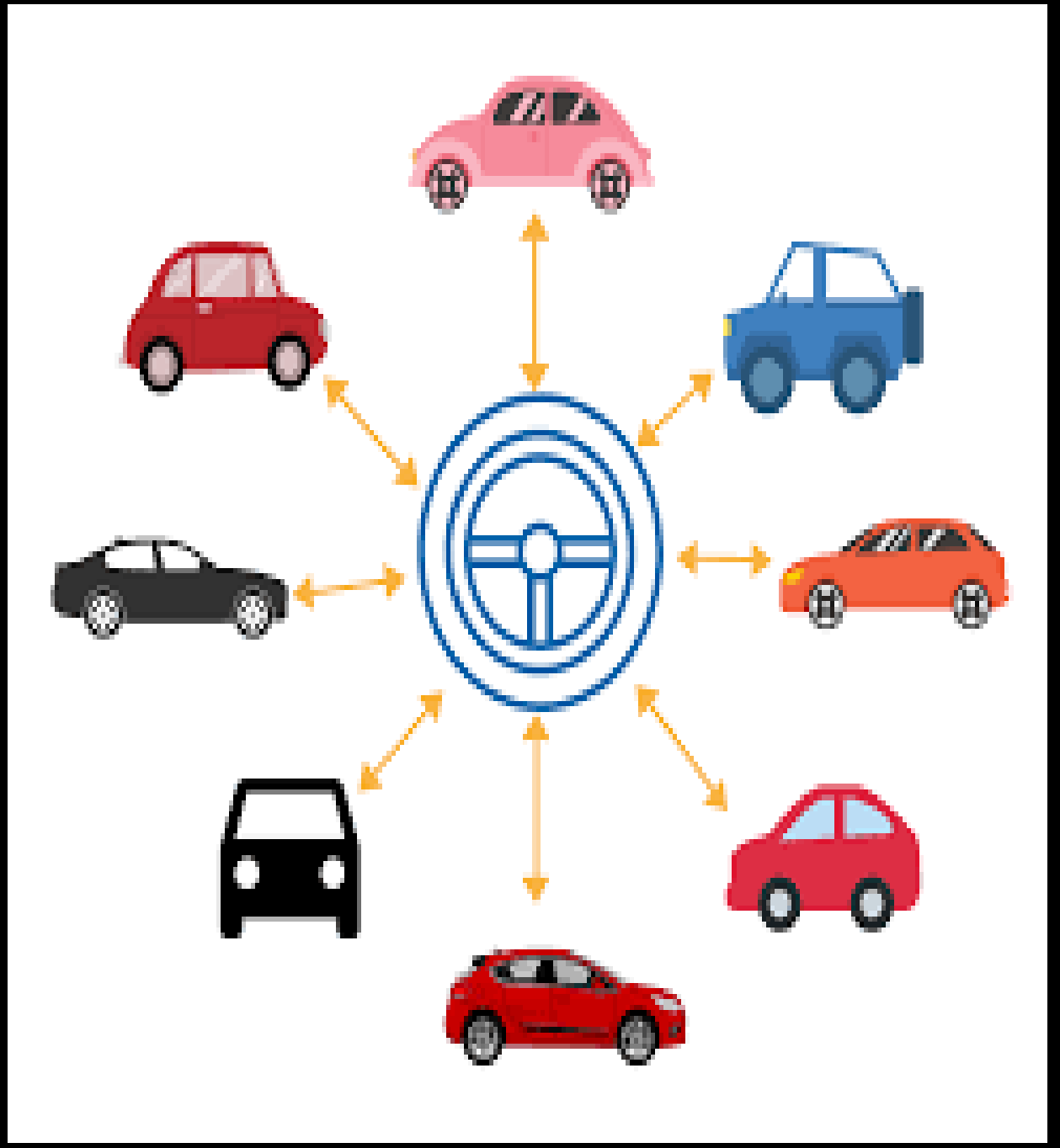
setSpeed() getSpeed()

drive()

attributes

fuel

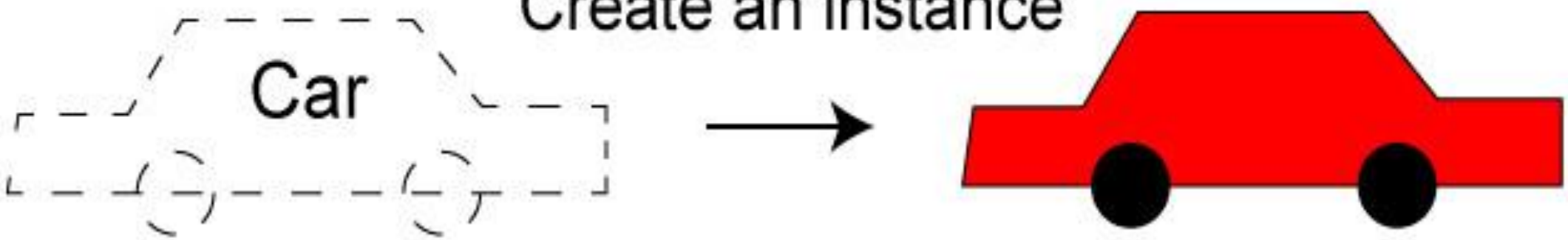
maxspeed



Class

Object

Create an instance



Properties	**Methods** - behaviors
color	start()
price	backward()
km	forward()
model	stop()

Property values	**Methods**
color: red	start()
price: 23,000	backward()
km: 1,200	forward()
model: Audi	stop()

Examples of Objects

LightBulb

- **state/attributes**
  - on (true or false)
- **behavior**
  - switch on
  - switch off
  - check if on

Car

- **state/attributes**
  - # of liters of gas in tank
  - total # of km run so far
  - efficiency (km/liter)
- **behavior**
  - drive
  - load gas
  - change efficiency
  - check gas
  - check odometer reading

BankAccount

- **state/attributes**
  - balance
- **behavior**
  - deposit
  - withdraw
  - check balance

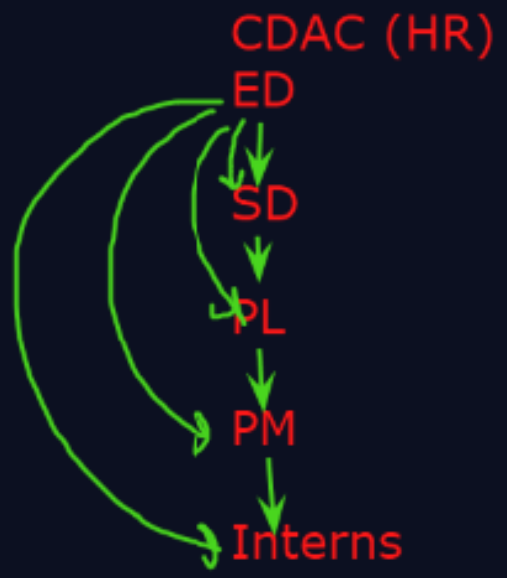
Note

- **each object is an "instance" of that "class" of object**
- **each instance has its own values for its attributes**
  - e.g., different accounts can have different balances

OOPS : Objected Oriented Programming:

-1. Modularity : Divides the program into objects.
-2. Resusability : Inherit existing functionality.
-3. Scalability : Easier to manage large appications.
-4. Secure : Abstraction and Encapsulation we restrict direct access to data

Data hiding : Access Modifiers (public,private, protected, defa



Real world Example:

-1. Class
-2. Object
-3. Methods

Key Features:

-1. Class
-2. Object
-3. Abstraction
-4. Encapsulation
-5. Inheritance
-6. Polymorphism

4 pillars → Major Pillars:Abstraction, Encapsulation, Modularity, Hiera

4 pillars → Minor Pillars:Typing, Concurrency, Persistence

Real world Example:

-1. Class
-2. Object
-3. Methods

Key Features:

-1. Class
-2. Object
    -Objects are real worlds entity, and classes are their blueprints.

```
class Student{
    int i=10;
    void display(){
    }
    p.s.v.main(){

    }
}
```

```
        show();//Function call
        SOP (show());//Direct printing od return data
        //or
        String x = show();
        SOP(x);
    }
}
```

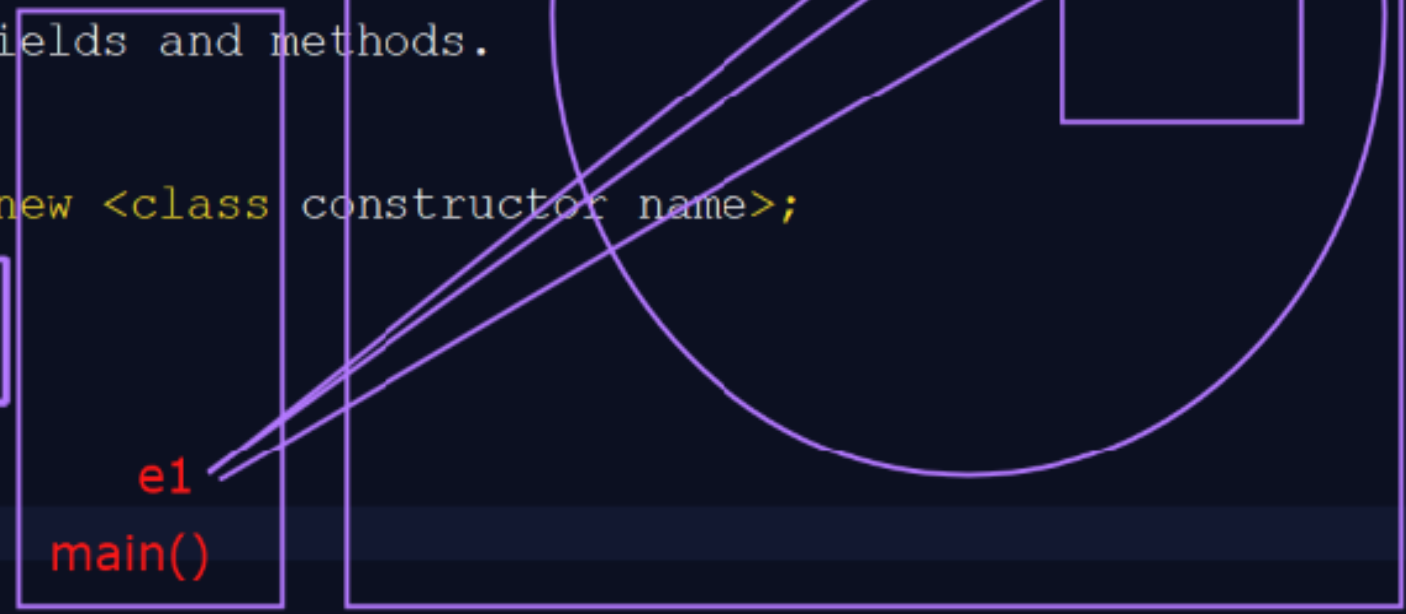
095_Rohan Sharma_KH raised hand View ×

Reference:
-A reference variable stores the memory address of an object.

-It is used to access the object'fields and methods.

Syntax:
    <Classname> <refernce name> = new <class constructor name>;

```
Employee e1 = new Employee();
```



```
    void getdata(){
        System.out.println("empId="+empid);
        System.out.println("empName="+empname);
    }

    void show(){
        System.out.println(empid+" "+empname);
    }


}

//Driver class
class EmployeeDemo{

    public static void main(String args[]){

        System.out.println("Employee detail
        Employee e1 = new Employee();
        System.out.println(e1.empid);
        System.out.println(e1.empname);

        e1.setdata(111,"Ajay");
        System.out.println(e1.empid);
        System.out.println(e1.empname);

        e1.getdata();

    }
```



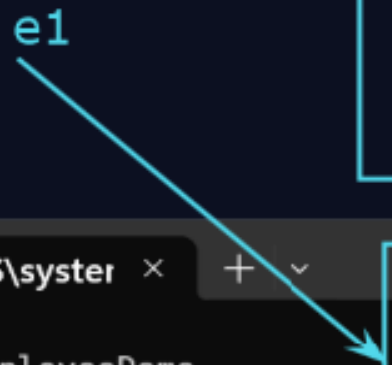
```
C:\WINDOWS\syster
                    System.out.println(e1.empname);

4 errors

C:\Test>javac EmployeeDemo.java

C:\Test>java EmployeeDemo
Employee details
0
null
111
Ajay
empId=111
empName=Ajay

C:\Test>
```

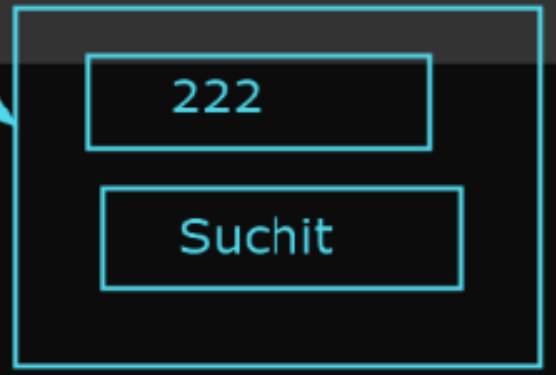
e1

111
Ajay

```java
//Driver class
class EmployeeDemo{

    public static void main(String args[]){

        System.out.println("Employee details");
        Employee e1 = new Employee();
        System.out.println(e1.empid);
        System.out.println(e1.empname);

        e1.setdata(111,"Ajay");
        System.out.println(e1.empid);
        System.out.println(e1.empname);
        e1.getdata();

        e1.setdata(222,"Suchit"); //Resettig of
        System.out.println(e1.empid);
        System.out.println(e1.empname);

        e1.getdata();

    }


}
```

```
C:\Test>java EmployeeDemo
Employee details
empId=100
100
null
111
Ajay
empId=111
empName=Ajay
222
Suchit
empId=222
empName=Suchit

C:\Test>
```





```java
//Driver class
class EmployeeDemo1{

    public static void main(String args[]){

        System.out.println("Employee details");
        Employee e1 = new Employee();
        //System.out.println(e1.empid);
        //System.out.println(e1.empname);

        System.out.println();
        e1.setdata(111,"Ajay");
        System.out.println(e1.empid);
        System.out.println(e1.empname);
        e1.getdata();
        System.out.println();

        Employee e2 = new Employee();
        e2.setdata(222,"Suchit"); //Resettig of
        System.out.println(e2.empid);
        System.out.println(e2.empname);

        e1.getdata();

    }
```



```
C:\WINDOWS\syster

C:\Test>java EmployeeDemo1
Employee details
empId=100

111
Ajay
empId=111
empName=Ajay

222
Suchit
empId=222
empName=Ajay

C:\Test>
```

```
    System.out.println("This is overloading: 3: x="+x1+"a1="+a1 );
}

public static void main(String args[]){


    MethodOverloadingDe
    //MEthod 1
    d1.show(); //compil
    d1.show(10); //comp

    //MEthod 2
    d1.show(); //compil
    d1.show(10);//compi
    d1.show(10.0f);//co

    //Method 3
    d1.show(12,3.45f);
    d1.show(3.45f, 55);

}
```

You are screen sharing HI Stop share

C:\WINDOWS\syster

```
                    ^
MethodOverloadingDemo.java:28: error: incompatible types: possible lossy convers
                x = id;
                    ^
MethodOverloadingDemo.java:35: error: incompatible types: possible lossy convers
                x = id;
                    ^
3 errors

C:\Test>javac MethodOverloadingDemo.java

C:\Test>java MethodOverloadingDemo
This is overloading: 1
This is overloading: 2: x=10
This is overloading: 1
This is overloading: 2: x=10
This is overloading: 3: x=10.0
This is overloading: 3: x=3.45a1=12
This is overloading: 3: x=3.45a=55

C:\Test>
```

```java
}

//Driver class
class CustomerDemo1{


    public static void main(String

        Customer c = new Customer(
        c.display();
        Customer c1 = new Customer
        c1.display();
        Customer c2 = new Customer("Komal",19,400.00);
        c2.display();
        Customer c3;
        c3 = new Customer("Samiksha",20);
        c3.display();

    }

}
```

You are screen sharing HI Stop share

C:\WINDOWS\syster

```
Name=KomalAge=19Cost=400.0
Name=SamikshaAge=20Cost=0.0

C:\Test>javac CustomerDemo1.java

C:\Test>java CustomerDemo1
Name=nullAge=0Cost=100.0
Name=ShrutiAge=18Cost=500.0
Name=KomalAge=19Cost=400.0
Name=SamikshaAge=20Cost=0.0

C:\Test>
```