Introduction to Class and Object

Class

- Class is **derived datatype**, it combines members of different datatypes into one.
- Defines new datatype (primitive ones are not enough).
  - For Example : **Car, College, Bus etc..**
- This new datatype can be used to create objects.
- A class is a template for an object .

Example :

```
class Car{
        String company;
        String model;
        double price;
        double milage;
        .........
}
```

Car Class

Class: Car

Properties (Describe)
Company
Model
Color
Mfg. Year
Price
Fuel Type
Mileage
Gear Type
Power Steering
Anti-Lock braking system

Methods (Functions)
Start
Drive
Park
On_break
On_lock
On_turn

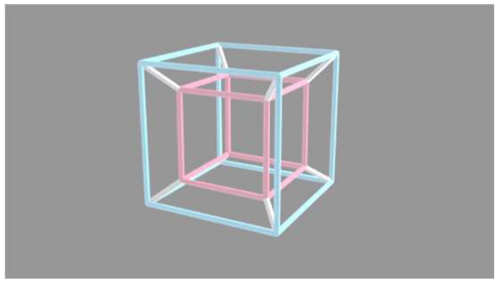
Object

- An object is an **instance** of a **class**.
- An object has a **state** and **behavior**.

  Example: A dog has

  states - color, name, breed as well as

  behaviors – barking, eating.
- The **state** of an object is stored in **fields** (variables), while **methods** (functions) display the object's **behavior**.



What is an Object?

Philosophy of Object Oriented

- Our real world is nothing but classification of objects
  - E.g. Human, Vehicle, Library, River, Watch, Fan, etc.
- Real world is organization of different objects which have their own characteristics, behavior
  - Characteristic of Human: Gender, Age, Height, Weight, Complexion, etc.
  - Behavior of Human: Walk, Eat, Work, React, etc.
  - Characteristic of Library: Books, Members, etc.
  - Behavior of Library: New Member, Issue Book, Return Book etc.
- The OO philosophy suggests that the things manipulated by the program should correspond to things in the real world.
  - **Classification** is called a Class in OOP
  - **Real world entity** is called an Object in OOP
  - **Characteristic** is called Property in OOP
  - **Behavior** is called Method in OOP

What is an Object?

What is an Object?

Pe
n

Board

Lapto
p

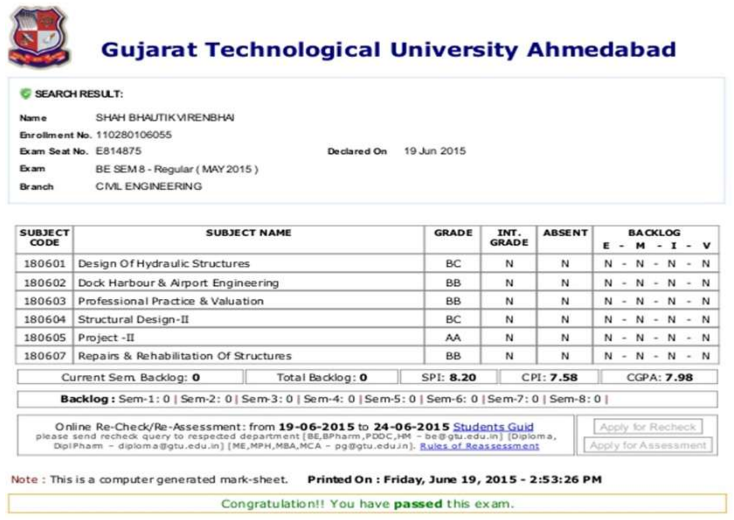
Bench

Projecto
r

Bik
e

Physical objects…

What is an Object? (Cont…)



Gujarat Technological University Ahmedabad

SEARCH RESULT:

Name	SHAH BHAUTIK VIRENBHAI		
Enrollment No.	110280106055		
Exam Seat No.	E814875	Declared On	19 Jun 2015
Exam	BE SEM 8 - Regular (MAY 2015)		
Branch	CIVIL ENGINEERING		

SUBJECT CODE	SUBJECT NAME	GRADE	INT. GRADE	ABSENT	BACKLOG E - M - I - V
180601	Design Of Hydraulic Structures	BC	N	N	N - N - N - N
180602	Dock Harbour & Airport Engineering	BB	N	N	N - N - N - N
180603	Professional Practice & Valuation	BB	N	N	N - N - N - N
180604	Structural Design-II	BC	N	N	N - N - N - N
180605	Project -II	AA	N	N	N - N - N - N
180607	Repairs & Rehabilitation Of Structures	BB	N	N	N - N - N - N

Current Sem. Backlog: **0**	Total Backlog: **0**	SPI: **8.20**	CPI: **7.58**	CGPA: **7.98**

Backlog : Sem-1: 0 | Sem-2: 0 | Sem-3: 0 | Sem-4: 0 | Sem-5: 0 | Sem-6: 0 | Sem-7: 0 | Sem-8: 0 |

Online Re-Check/Re-Assessment: from **19-06-2015** to **24-06-2015** Students Guid
please send recheck query to respected department [BE,BPharm,PDDC,HM - be@gtu.edu.in] [Diploma, DiplPharm - diploma@gtu.edu.in] [ME,MPH,MBA,MCA - pg@gtu.edu.in]. Rules of Reassessment

Apply for Recheck
Apply for Assessment

Note : This is a computer generated mark-sheet. **Printed On : Friday, June 19, 2015 - 2:53:26 PM**

Congratulation!! You have **passed** this exam.

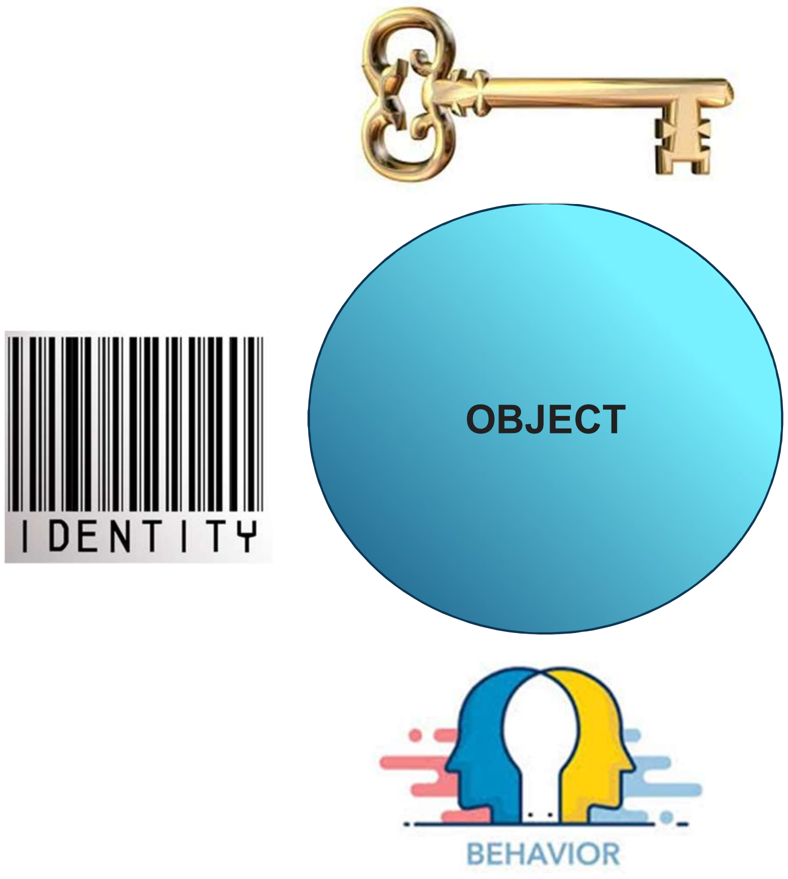
Result

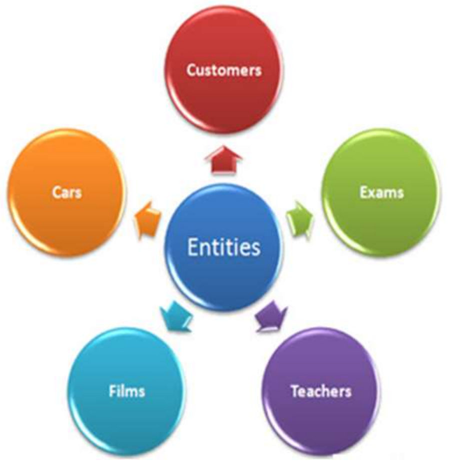
Bank
Account

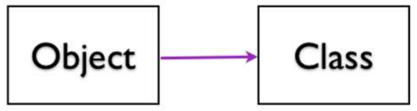
Logical objects…

What is an Object?

- An Object is a key to understand Object Oriented Technology.
- An entity that has state and behavior is known as an object. e.g., Mobile, Car, Door, Laptop etc
- Each and every object posses

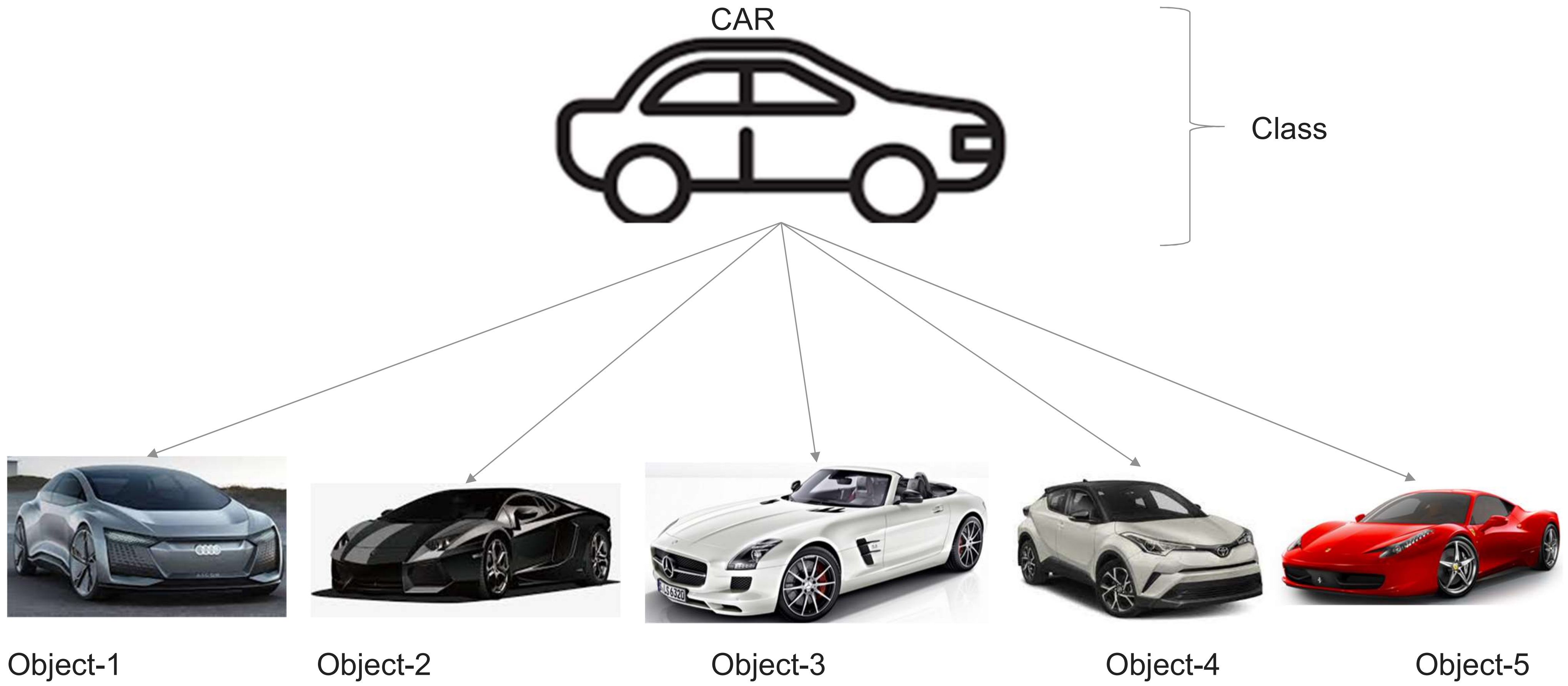




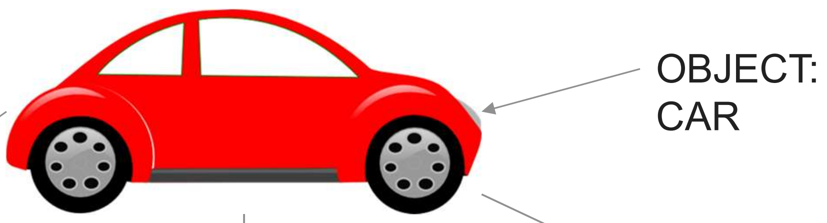


Object is an Instance of Class

Object: A Real-World Entity



Object: A Real-World Entity

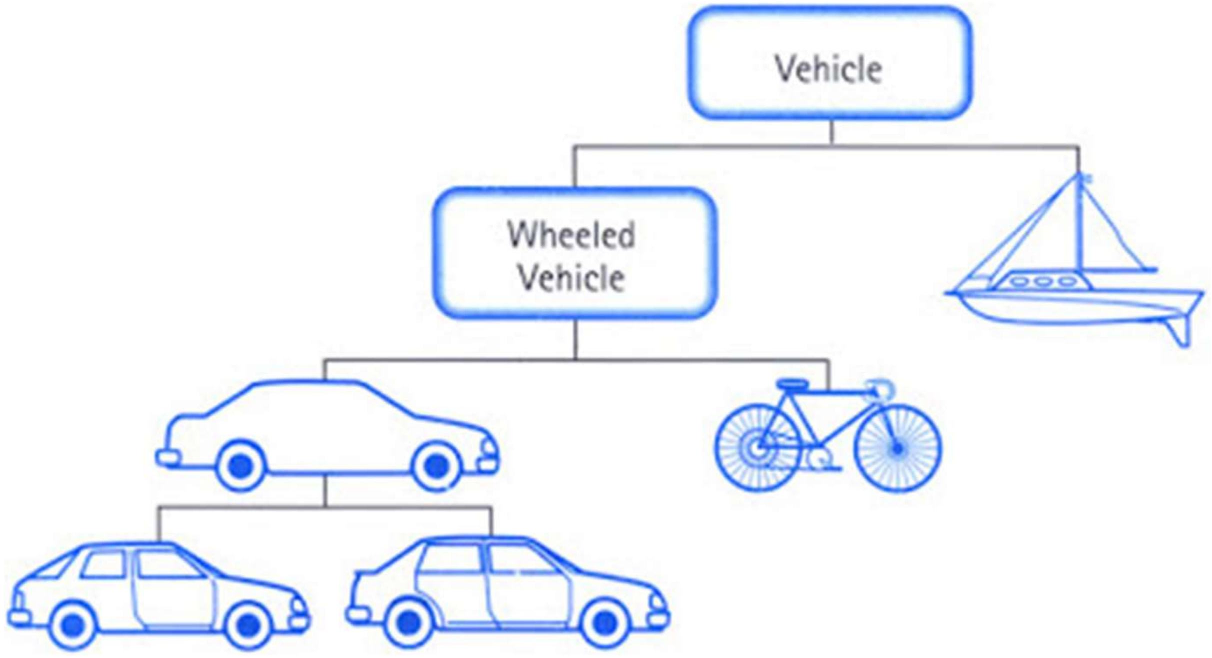


Properties (Describe)
Manufacturer
Model
Color
Year
Price

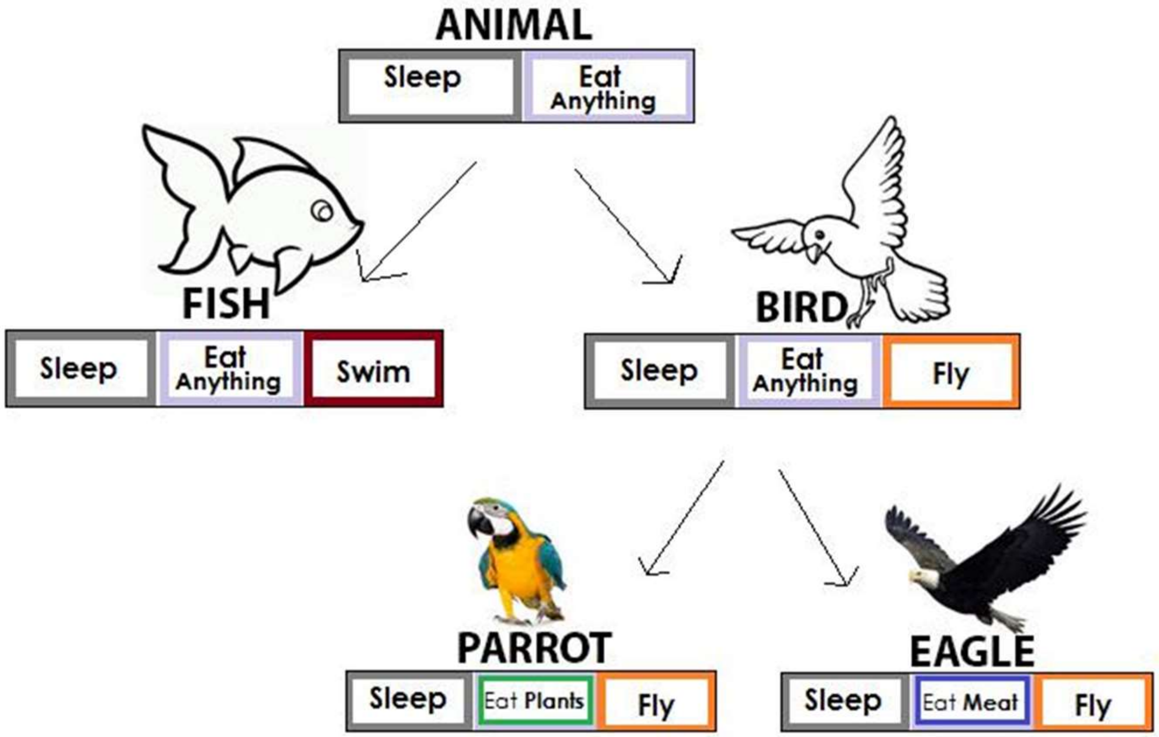
Methods (Actions)
Start
Drive
Park

Events
On_Start
On_Parked
On_Brake

Object: A Real-World Entity



Object: A Real-World Entity



Objects of Class Bird

Classes and Objects

Classes and Objects

Class

Object 1

Object 2

Blueprint

Houses built according to the blueprint

Class is a blueprint of an object

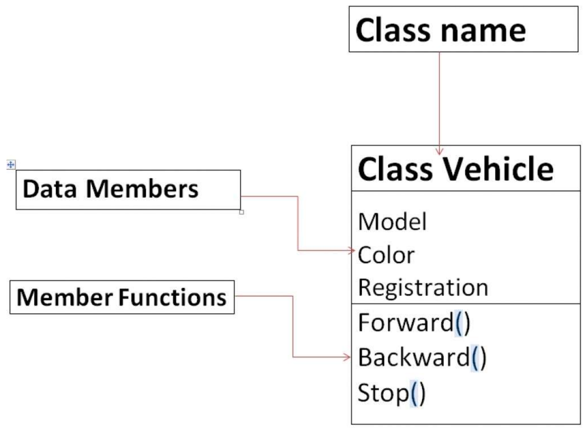
Class describes the object

Object is instance of class

What is Class?

- **Class can be defined in multiple ways**
  - A class is the building block.
  - A class is a blueprint for an object.
  - A class is a user-defined data type.
  - A class is a collection of objects of the similar kind.
  - A class is a user-defined data type which combines data and methods.
  - A class describes both the data and behaviors of objects.
- Class contains data members (also known as field or property or data) and member functions (also known as method or action or behavior)
- Classes are similar to structures in C.
- Class name can be given as per the Identifier Naming Conventions.

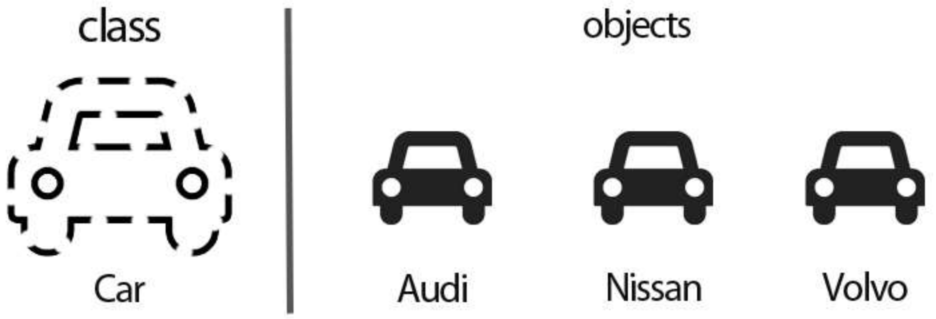
Blueprint
Houses built according to the blueprint



What is Object?

- **Definition**: An Object is an instance of a Class.
- An Object is a variable of a specific Class
- An Object is a data structure that encapsulates data and functions in a single construct.
- Object is a basic run-time entity
- Objects are analogous to the real-w entities.

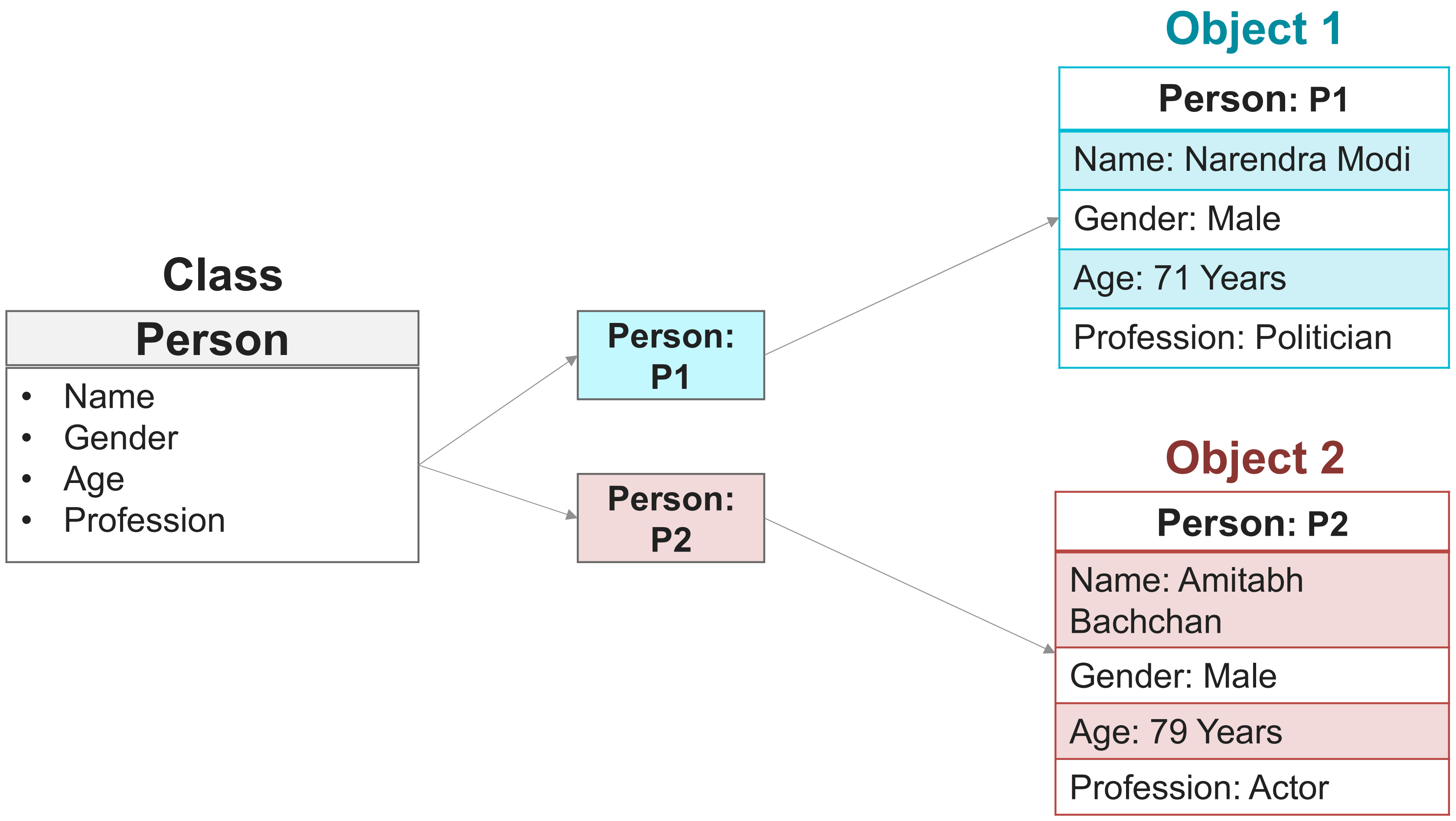
Blueprint
Houses built according to the blueprint



Points to Remember

- When a class is defined, only the specification or blueprint for the object is defined; no memory or storage is allocated.
- When an object of a class is declared, the memory is allocated as per the data members of a class
- We can access the data members and member functions of a class by using a . (dot) operator.
- Generally Class contains
    - Data Members
    - Member Functions
    - Constructor (Special Member Function)

Class and Objects



Class

Person
• Name • Gender • Age • Profession

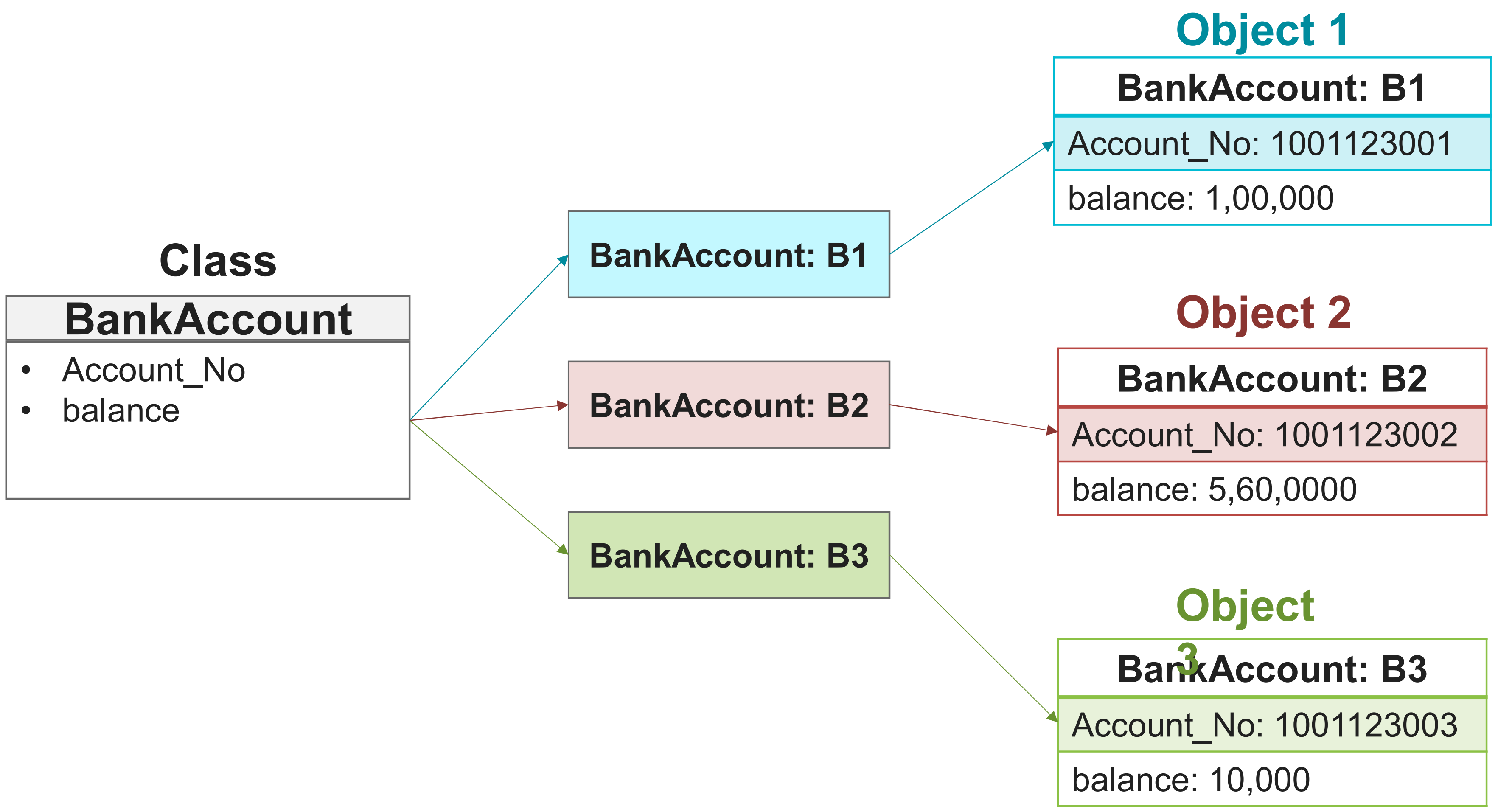
Object 1

Person: P1
Name: Narendra Modi
Gender: Male
Age: 71 Years
Profession: Politician

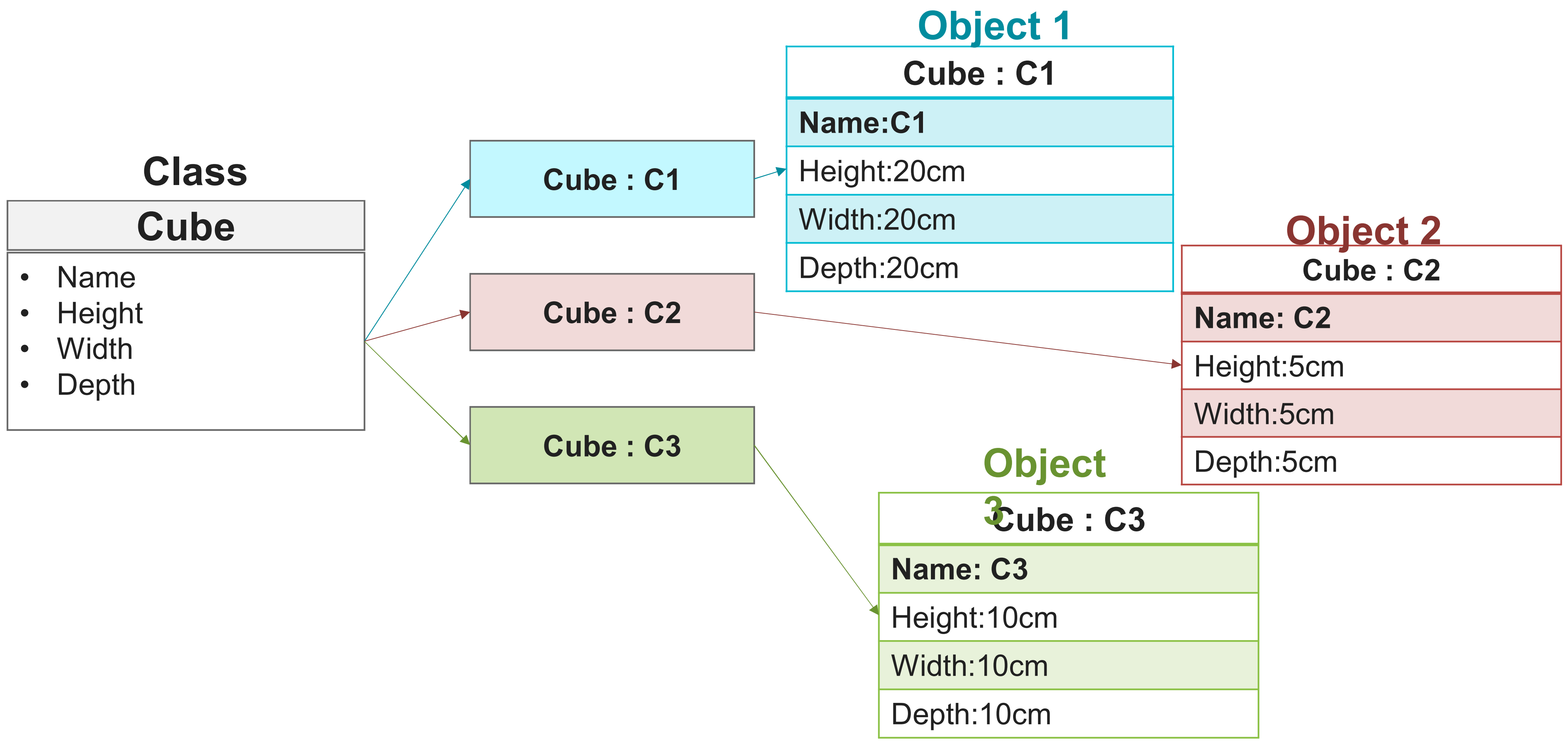
Object 2

Person: P2
Name: Amitabh Bachchan
Gender: Male
Age: 79 Years
Profession: Actor

Class and Objects



Class and Objects



Creating Object & Accessing members

- **new** keyword creates new object
- Syntax:

```
ClassName      objName = new      ClassName();
```

Example :

```
SmartPhone     iPhone = new       SmartPhone();
```

- Object variables and methods can be accessed using the **dot (.)** operator
- Example:

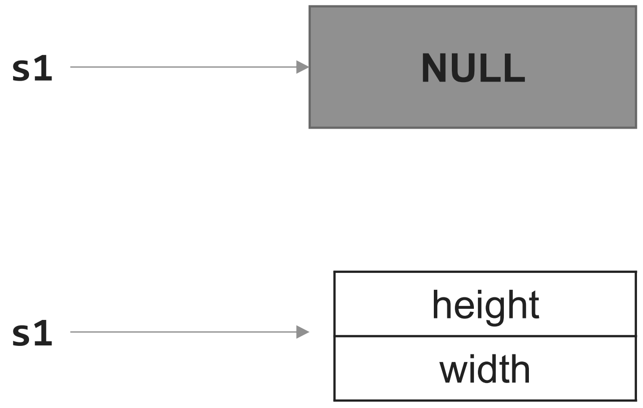
```
iPhone.storage = 8000;
```

Declaring an Object

- When we create a class, we are creating a **new data type**.
- Object of that data type will have all the attributes and abilities that are designed in the class

MyProg.java

```
class Square{
      double height;
      double width;
}
class MyProg{
      public static void main(String[] args)
      {
         Square s1=new Square();
      }
}
```



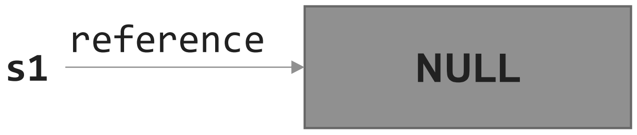
- The new operator dynamically allocates (that is, allocates at run time) memory for an object and returns a reference to it.
- This reference is, more or less, the address in memory of the object allocated by new.
- This reference is then stored in the variable. Thus, in Java, all class objects must be dynamically allocated.

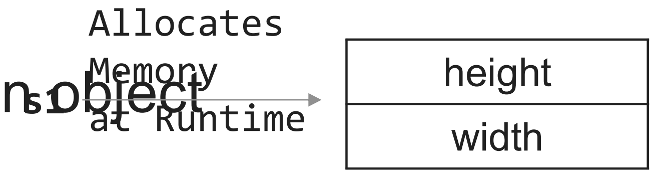
Declaring an Object

MyProg.java

```
class Square{
    double height;
    double width;
}
class MyProg{
    public static void main(String[] args) {
        Square s1; //declare reference to object
        s1= new Square();//allocate a Square object
  }
}
```

An object reference is similar to a memory pointer.



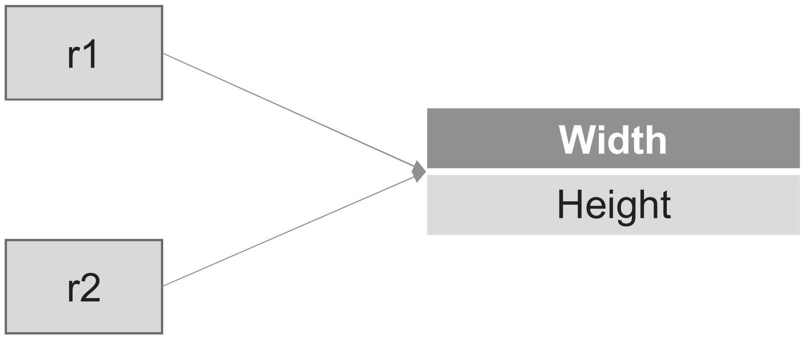


- **new** operator dynamically allocates memory for an object
- Here, s1 is a variable of the class type.
- The class name followed by parentheses specifies the constructor for the class.
- It is important to understand that **new** allocates memory for an object during run time.

Assigning Object Reference

ObjectDemo.java

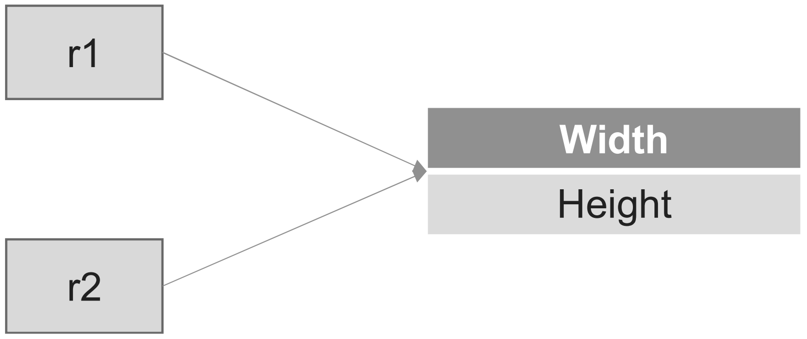
```
Rectangle r1=new Rectangle();
Rectangle r2=r1;
```



Here, **r1** and **r2** will both refer to the *same* object. The assignment of **r1** to **r2** did not allocate any memory or copy any part of the original object. It simply makes **r2** refer to the same object as does **r1**.

ObjectDemo.java

```
Rectangle r1=new Rectangle();
Rectangle r2=r1;
…
R1=null
```



Here, **r1** has been set to **null**, but **r2** still points to the original object.

WAP using class Person to display name and age

```java
class MyProgram {
public static void main(String[] args)
  {
    Person p1= new Person();
    Person p2= new Person();
    p1.name="modi";
    p1.age=71;
    p2.name="bachchan";
    p2.age=80;
  System.out.println("p1.name="+p1.name);
  System.out.println("p2.name="+p2.name);
    System.out.println("p1.age="+p1.age);
    System.out.println("p2.age="+p2.age);
  }//main()
}//class myProgram
```

```java
class Person
{
    String name;
    int age;
}//class person
```

Output

```
p1.name=modi
p2.name=bachchan
p1.age=71
p2.age=80
```

WAP using class Person to display name and age with method

```java
class MyProgram {
public static void
        main(String[] args){
    Person p1=new Person();
    Person p2=new Person();
    p1.name="modi";
    p1.age=71;
    p2.name="bachchan";
    p2.age=80;
    p1.displayName();
    p2.displayName();
    p1.displayAge();
    p2.displayAge();
  } //main()
} //class myProgram
```

```java
class Person{
   String name;
   int age;
public void displayName(){
 System.out.println("name="+name);
 }
public void displayAge(){
  System.out.println("age="+age);
 }
}//class person
```

Output

```
name=modi
name=bachchan
age=71
age=80
```

WAP using class Rectangle and calculate area using method

```java
import java.util.*;
class MyProgram {
public static void main(String[] args){
    Rectangle r1=new Rectangle();
    Scanner sc=new Scanner(System.in);
    System.out.print("enter height:");
    r1.height=sc.nextFloat();
    System.out.print("enter width:");
    r1.width=sc.nextFloat();
    r1.calArea();
} //main()
}//class myProgram
```

```java
class Rectangle{
float height;
float width;
public void calArea()
{
System.out.println(
"Area="+height*width);
} //calArea()
} //class
```

Output

```
enter height:30.55
enter width:20.44
Area=624.442
```

WAP using class Rectangle and calculate area with Return value

```java
import java.util.*;
class MyProgram {
public static void main(String[] args){
    float area;
    Rectangle r1=new Rectangle();
    Scanner sc=new Scanner(System.in);
    System.out.print("enter height:");
    r1.height=sc.nextFloat();
    System.out.print("enter width:");
    r1.width=sc.nextFloat();
    area=r1.calArea();
    System.out.println(“Area="+area);
  }//main()
}//class myProgram
```

```java
class Rectangle{
float height;
float width;
public float calArea(){
    return height*width;
  }//calArea()
}//class
```

Output

```
enter height:30.55
enter width:20.44
Area=624.442
```

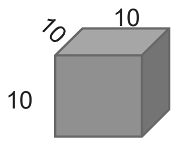
WAP using class Cube and calculate area using method with parameter

```
import java.util.*;
class MyProgramCube {
public static void main
                    (String[] args){
    float area;
    Cube c1= new Cube();
    area=c1.calArea(10,10,10);
System.out.println("area="+area);
  }//main()
}//class myProgram
```

```
class Cube{
float height;
float width;
float depth;
float calArea(float h, float
                                w, float d)
{    height=h;
     width=w;
     depth=d;
   return height*width*depth;
 }//calArea()
}//class
```

Output

```
area=1000.0
```



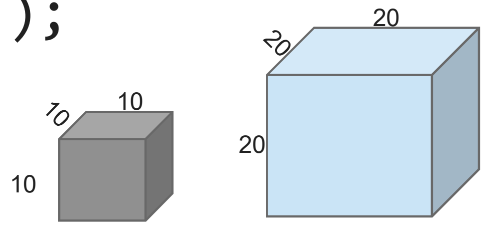
WAP using class Cube and calculate area of two objects

```java
import java.util.*;
class MyProgramCube {
public static void main
                 (String[] args){
  float area;
  Cube c1= new Cube(); //Obj1
  Cube c2= new Cube(); //Obj2
System.out.println("c1 area="

  +c1.calArea(10,10,10));
System.out.println("c2 area="

  +c2.calArea(20,20,20));
 } //main()
} //class
```

```java
class Cube{
float height;
float width;
float depth;
float calArea(float h, float
                          w, float d)
{   height=h;
    width=w;
    depth=d;
   return height*width*depth;
 } //calArea()
} //class
```



Output

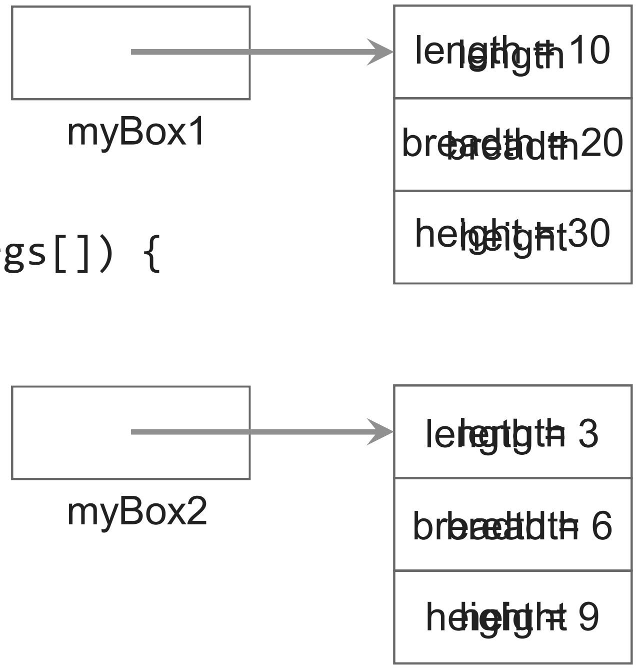
```
c1 area=1000.0
c2 area=8000.0
```

```
class Box {
       double length;
       double breadth;
       double height;
}
class BoxDemo {
       public static void main(String args[]) {
              Box myBox1 = new Box();
              Box myBox2 = new Box();
              double vol;

              myBox1.length = 10;
              myBox1.breadth = 20;
              myBox1.height = 30;

              myBox2.length = 3;
              myBox2.breadth = 6;
              myBox2.height = 9;

              vol = myBox1.length * myBox1.breadth * myBox1.height;
              System.out.println("Volume is " + vol);
              vol = myBox2.length * myBox2.breadth * myBox2.height;
              System.out.println("Volume is " + vol);
       }
}
```


myBox1
length = 10
breadth = 20
height = 30
myBox2
length = 3
breadth = 6
height = 9