Marwadi University
Marwadi Chandarana Group

Object-Oriented Programming

(01CT0105)

Objects and Classes

Prof. Suhag Baldaniya

Department of ICT

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

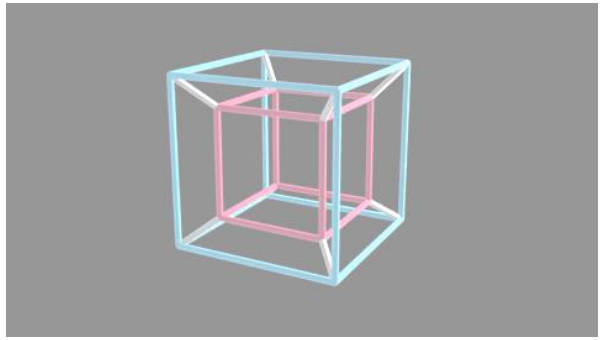
- An object is an **instance** of a **class**.
- An object has a **state** and **behavior**.

  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?

Pen

Board

Laptop

Bench

Projector

Bike

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, DipPharm - 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.

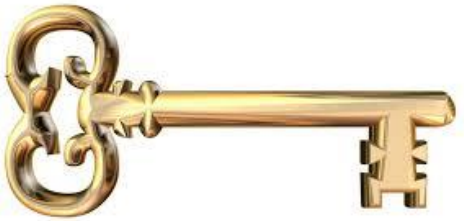
Result

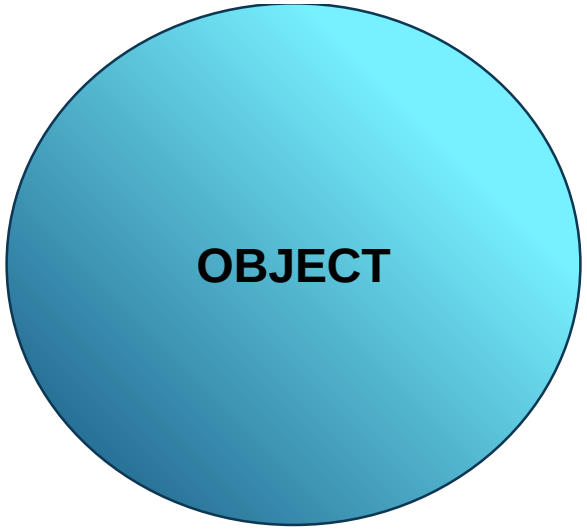
Bank Account

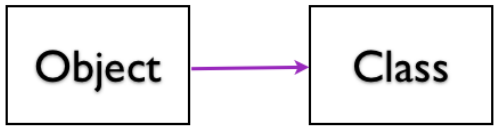
Logical objects…

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
  - Identity
  - State
  - Behavior

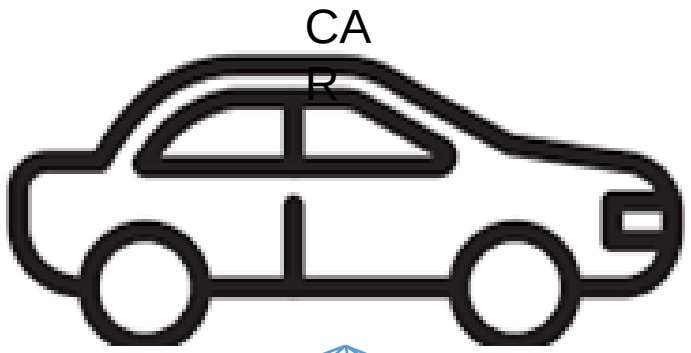






Object is an Instance of Class

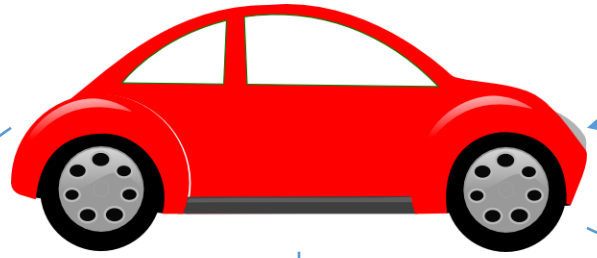
Object: A Real-World Entity



Clas
s

Object-1 Object-2 Object-3 Object-4 Object-5

Object: A Real-World Entity



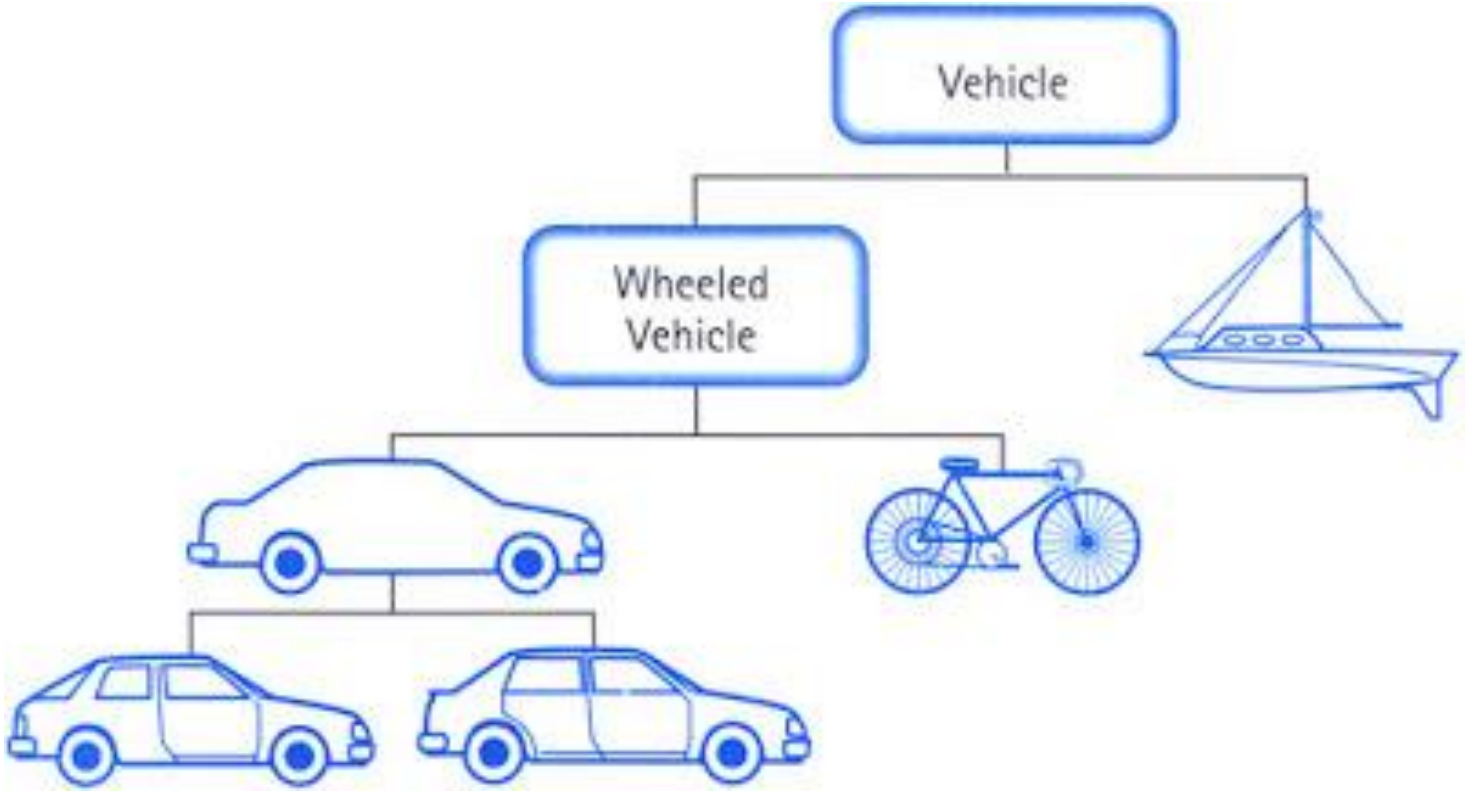
OBJECT:
CAR

Properties (Describe)
Manufacturer
Model
Color
Year
Price

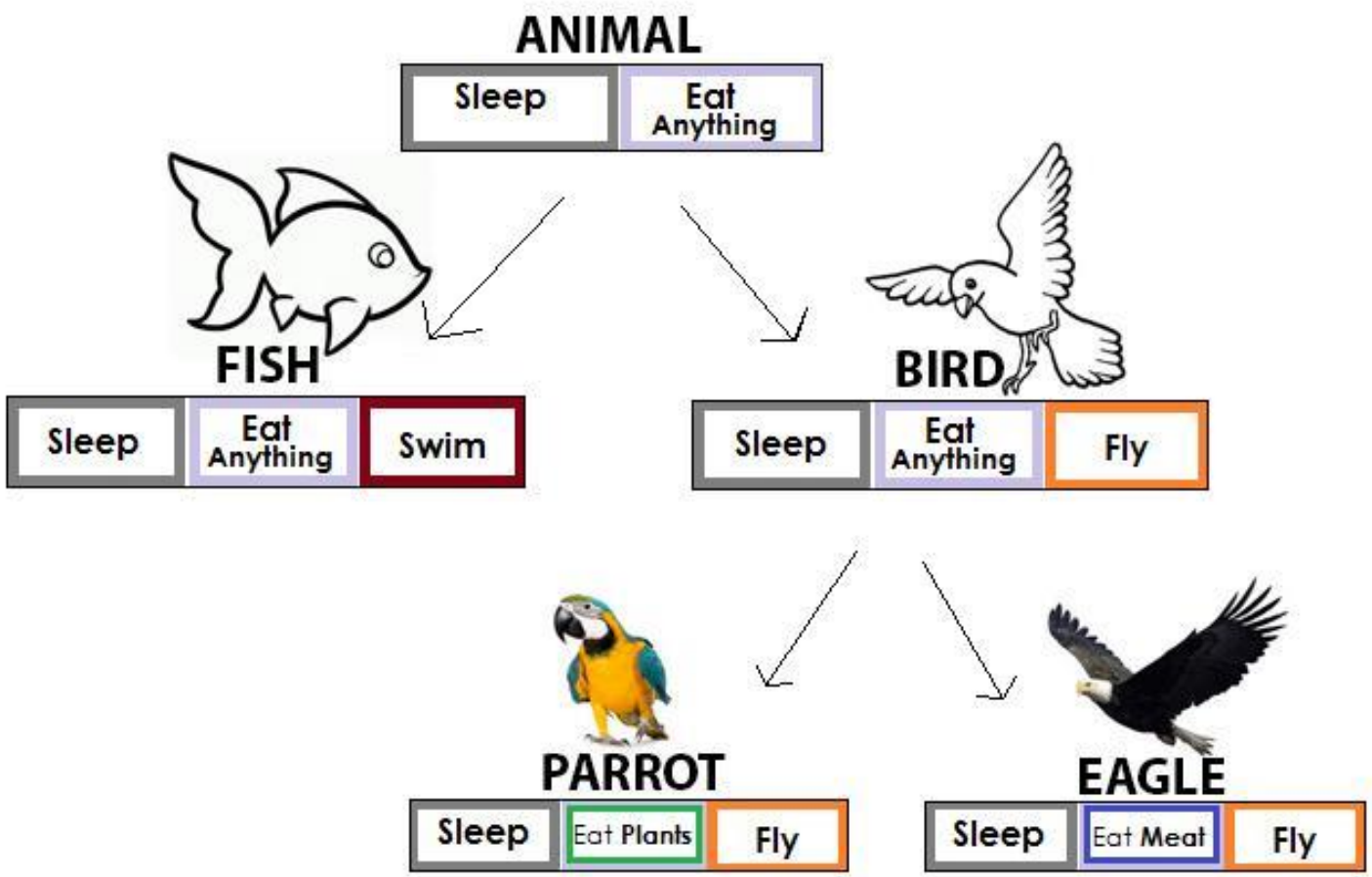
Methods (Actions)
Start
Drive
Park

Events
On_Start
On_Parked
On_Brake

Object: A Real-World Entity



Object: A Real-World Entity



Objects of Class Bird

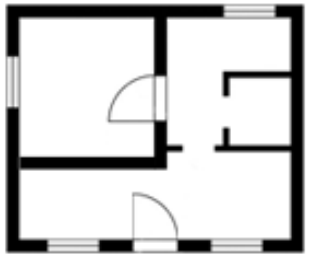




Classes and Objects

Classes and Objects

Class



Blueprint

Class is a blueprint of an object

Class describes the object

Object1

Object2

Houses built according to the blueprint

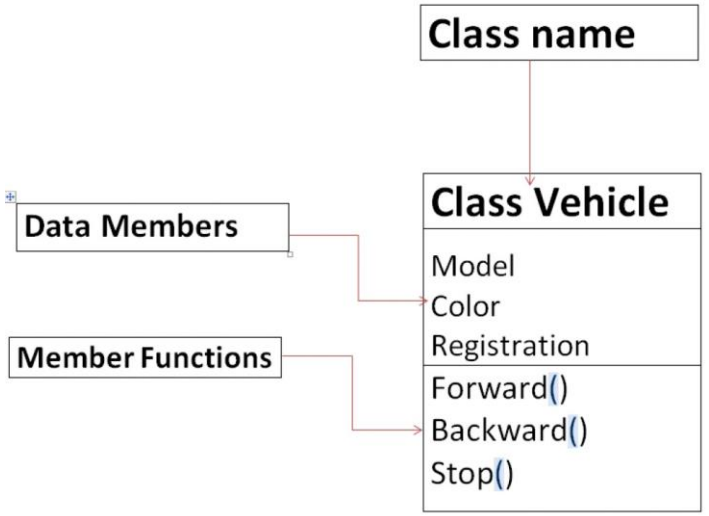
Object is instance of class

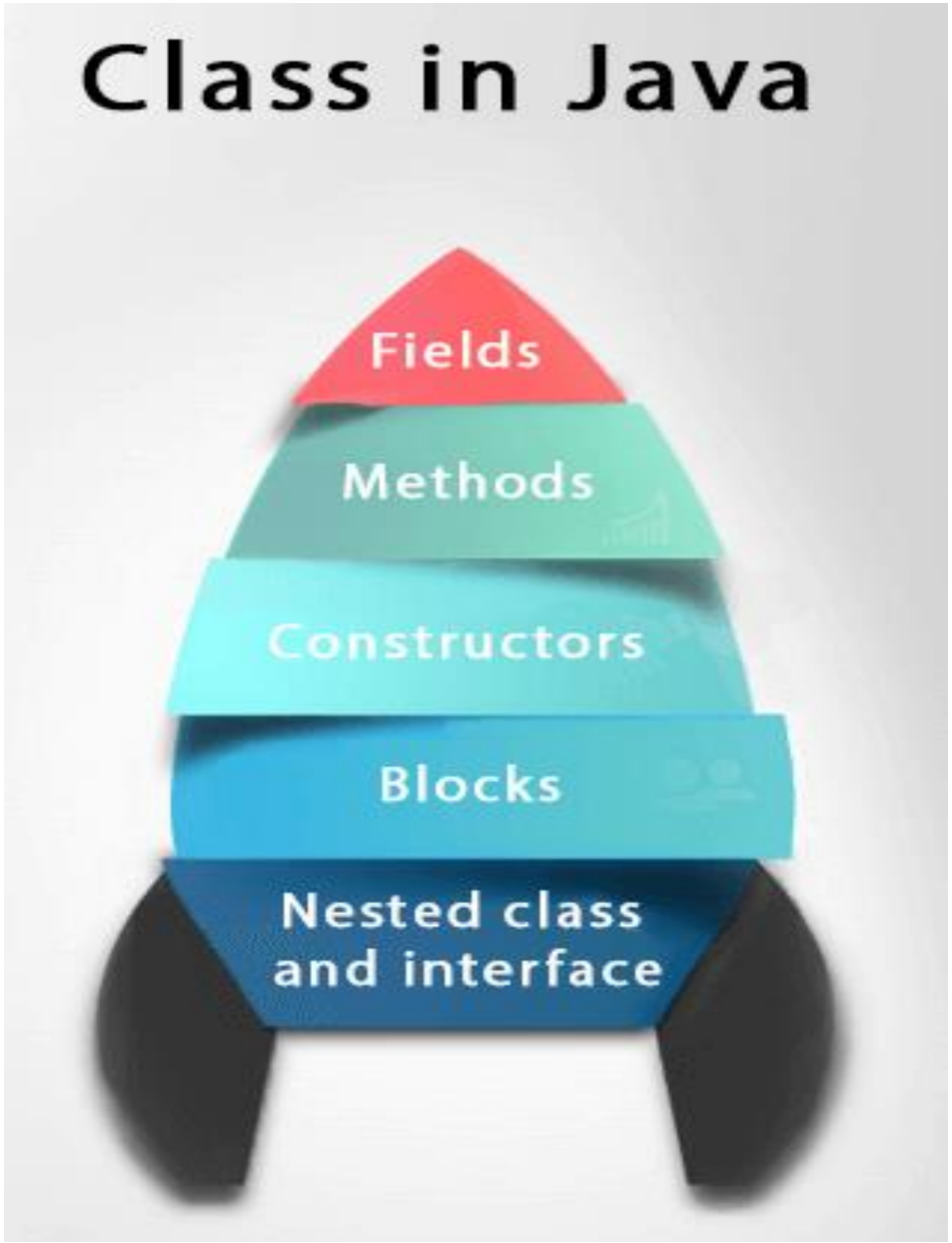
What is Class?

- **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





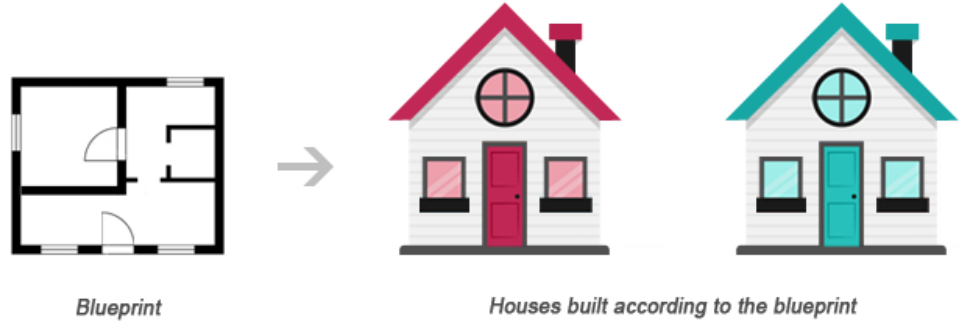
Syntax to declare a class:

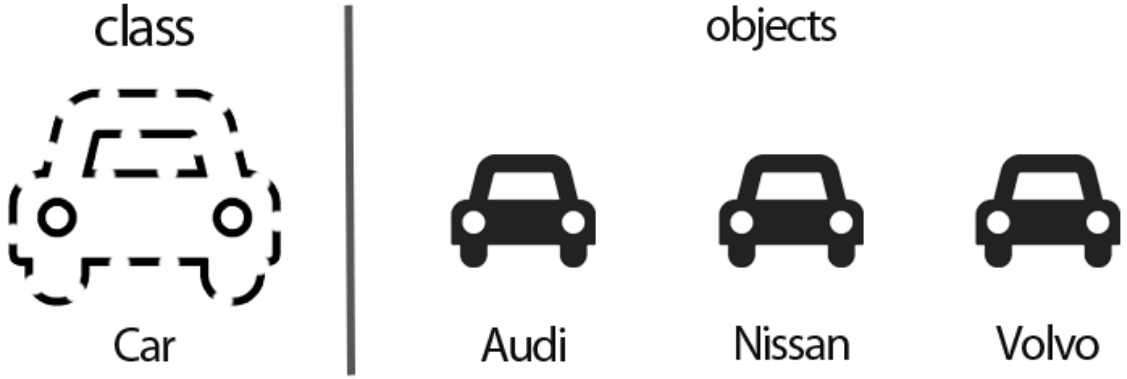
```
class <class_name>{
    field;
    method;
}
```

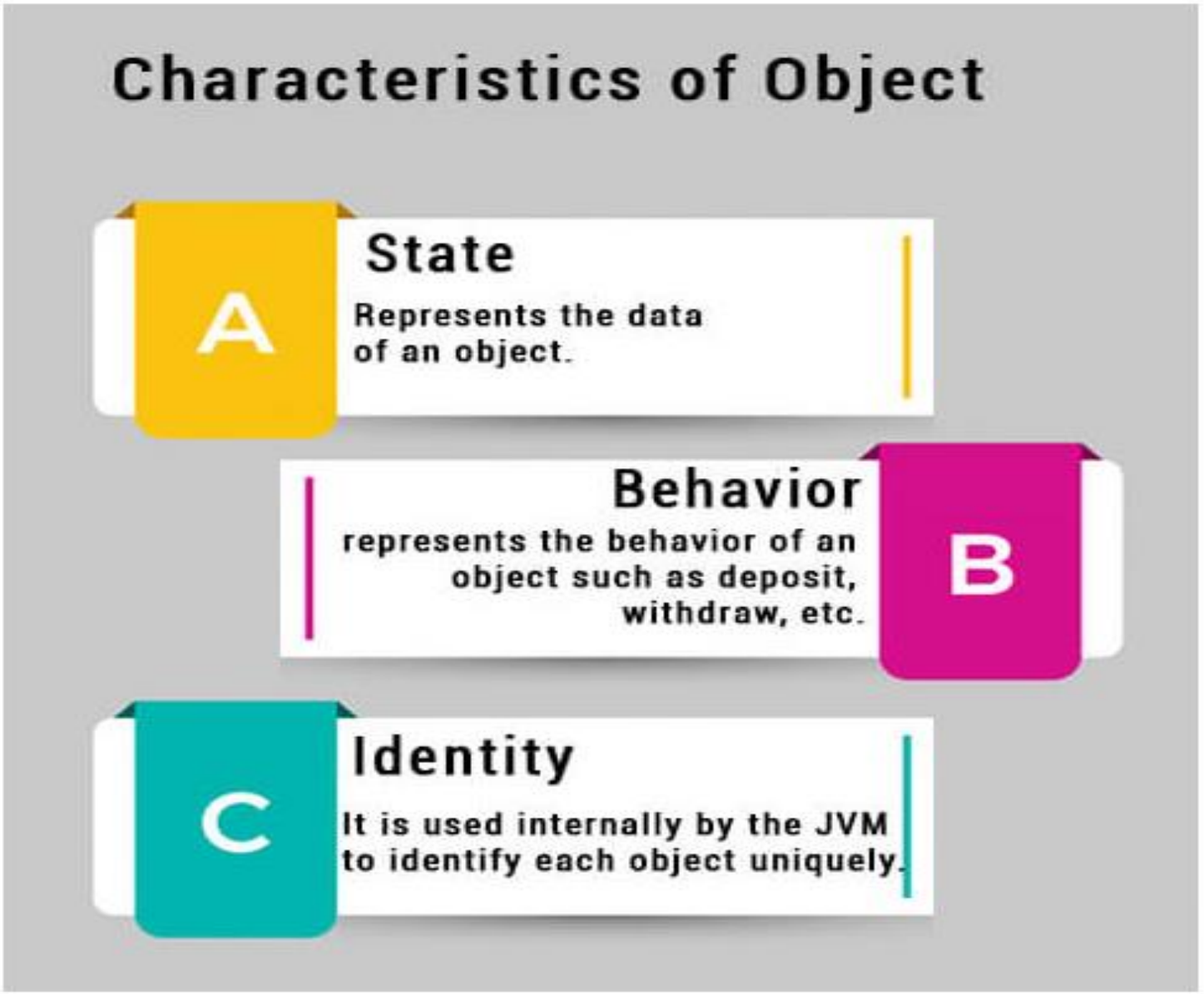
```
//Java Program to illustrate how to define a class and fields
1.//Defining a Student class.
class Student{
2.//defining fields
 int id;
 String name;
3.//creating main method inside the Student class
 public static void main(String args[]){
 4. //Creating an object or instance
  Student s1=new Student();//creating an object of Student
 5. //Printing values of the object
  System.out.println(s1.id);//accessing member through reference variable
  System.out.println(s1.name);
 }
}
```

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-world entities.






Characteristics of Object
A
State
Represents the data of an object.
Behavior
represents the behavior of an object such as deposit, withdraw, etc.
B
C
Identity
It is used internally by the JVM to identify each object uniquely.

Create Object

```
public class Main {
  int x = 5;

  public static void main(String[] args) {
    Main myObj = new Main();
    System.out.println(myObj.x);
  }
}
```

```
public class Main {
  int x = 5;

  public static void main(String[] args) {
    Main myObj1 = new Main();  // Object 1
    Main myObj2 = new Main();  // Object 2
    System.out.println(myObj1.x);
    System.out.println(myObj2.x);
  }
}
```

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)

Difference between Class & Object

Class	Object
Class is used as a template for declaring and creating the objects.	An object is an instance of a class.
When a class is created, no memory is allocated.	Objects are allocated memory space whenever they are created.
The class has to be declared first and only once.	An object is created many times as per requirement.
A class can not be manipulated as they are not available in the memory.	Objects can be manipulated.

Difference between Class & Object

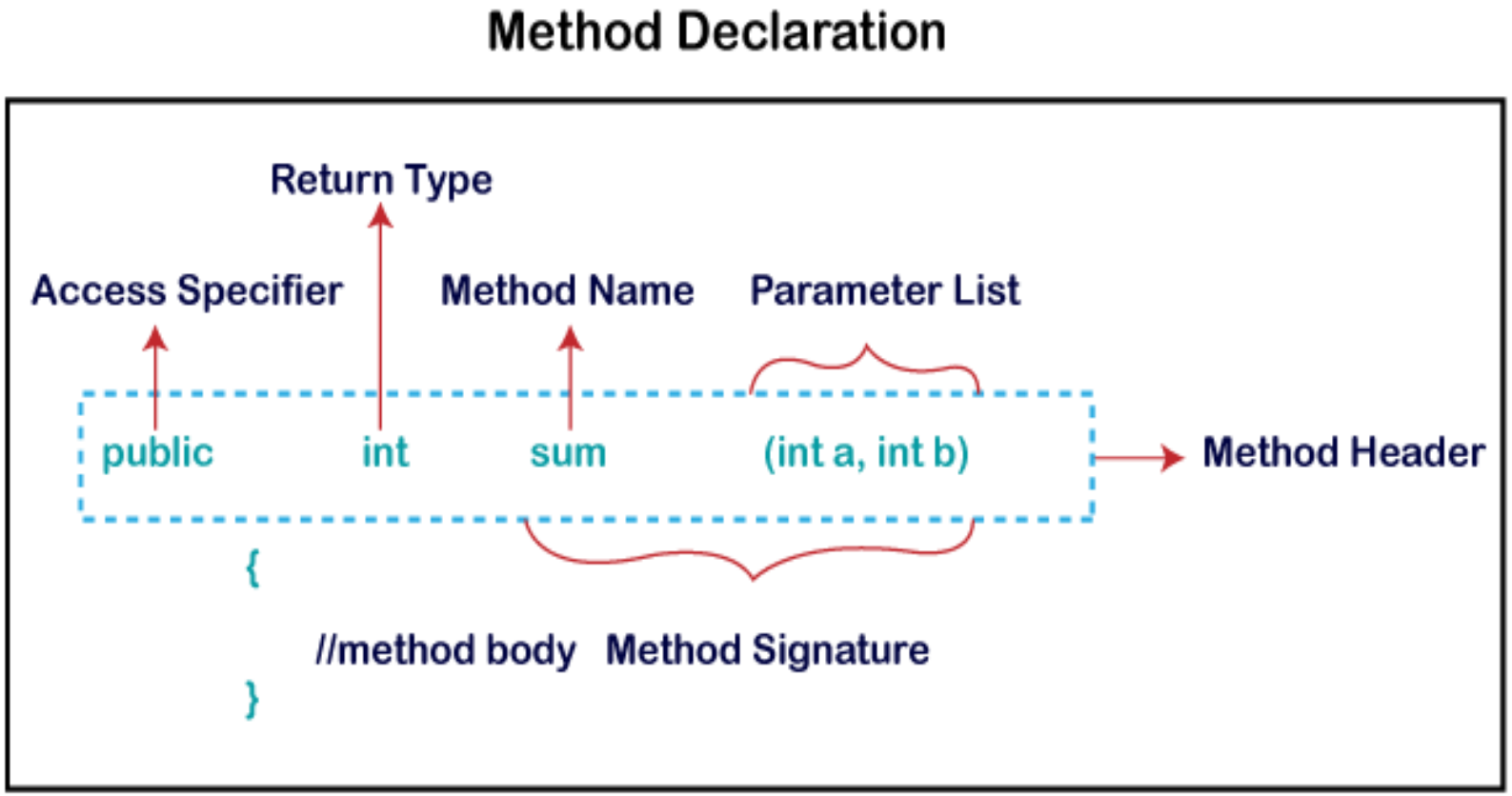
A class is a logical entity.	An object is a physical entity.
It is declared with the class keyword	It is created with a class name in C++ and with the **new** keywords in Java.
Class does not contain any values which can be associated with the field.	Each object has its own values, which are associated with it.
A class is used to bind data as well as methods together as a single unit.	Objects are like a variable of the class.

Method

- A **method** is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation.
- It is used to achieve the **reusability** of code.
- Do not require to write code again and again.
- It also provides the **easy modification** and **readability** of code, just by adding or removing a chunk of code.
- The method is executed only when we call or invoke it.

Method

- The method declaration provides information about method attributes, such as visibility, return-type, name, and arguments.
- It has six components that are known as **method header**, as we have shown in the following figure.



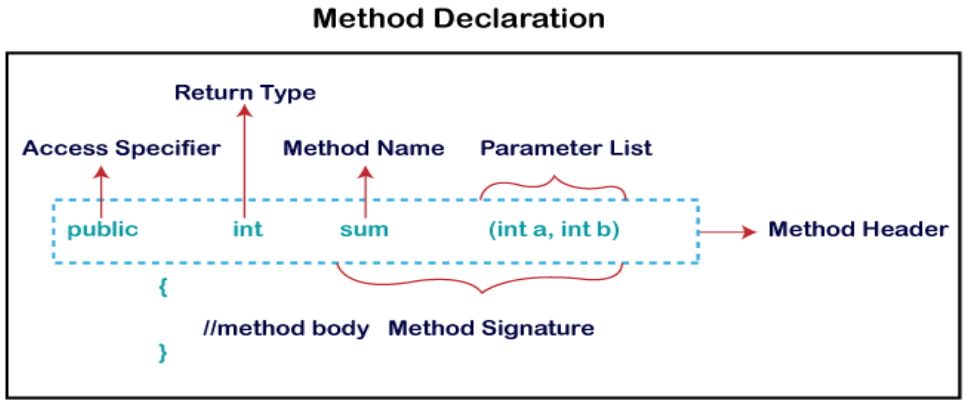
Method Header

- **Access Specifier:** Access specifier or modifier is the access type of the method.
- It specifies the visibility of the method. Java provides **four** types of access specifier:
- **Public:** The method is accessible by all classes when we use public specifier in our application.
- **Private:** When we use a private access specifier, the method is accessible only in the classes in which it is defined.
- **Protected:** When we use protected access specifier, the method is accessible within the same package or subclasses in a different package.
- **Default:** When we do not use any access specifier in the method declaration, Java uses default access specifier by default. It is visible only from the same package only.

Method

- **Return Type:** Return type is a data type that the method returns.
- It may have a primitive data type, object, collection, void, etc.
- If the method does not return anything, we use void keyword.

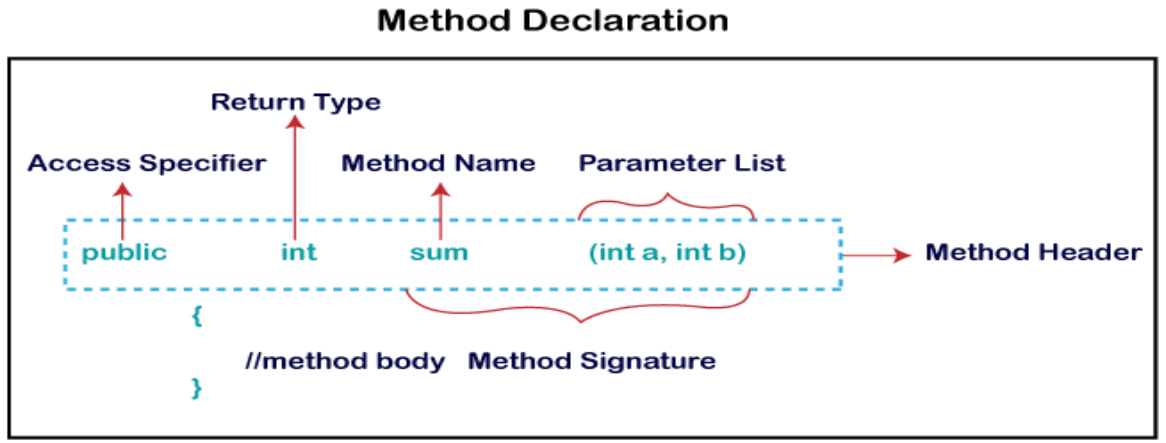
- **Method Name:** It is a unique name that is used to define the name of a method.
- It must be corresponding to the functionality of the method.
- Suppose, if we are creating a method for subtraction of two numbers, the method name must be **subtraction().**
- A method is invoked by its name.



Method

- **Parameter List:** It is the list of parameters separated by a comma and enclosed in the pair of parentheses.
- It contains the data type and variable name.
- If the method has no parameter, left the parentheses blank.

- **Method Body:** It is a part of the method declaration.
- It contains all the actions to be performed.
- It is enclosed within the pair of curly braces.



Types of Method

- There are two types of methods in Java:
- Predefined Method
- predefined methods are the method that is already defined in the Java class libraries is known as predefined methods.
- It is also known as the **standard library method** or **built-in method**.
- Example: **length(), equals(), compareTo(), sqrt(),** etc.

- User-defined Method
- The method written by the user or programmer is known as **a user-defined** method.
- These methods are modified according to the requirement.

Method

```
import java.util.Scanner;
public class EvenOdd
{
public static void main (String args[])
{
//creating Scanner class object
Scanner scan=new Scanner(System.in);
System.out.print("Enter the number: ");
//reading value from user
int num=scan.nextInt();
//method calling
findEvenOdd(num);
}
```

```
public static void findEvenOdd(int num)
{
//method body
if(num%2==0)
System.out.println(num+" is even");
else
System.out.println(num+" is odd");
}
}
```

Method

```
public class Addition
{
public static void main(String[] args)
{
int a = 19;
int b = 5;
//method calling
int c = add(a, b);   //a and b are actual parameters
System.out.println("The sum of a and b is= " + c);
}
```

```
//user defined method
public static int add(int n1, int n2)
//n1 and n2 are formal parameters
{
int s;
s=n1+n2;
return s; //returning the sum
}
}
```

Static Method

- A method that has static keyword is known as static method.
- In other words, a method that belongs to a class rather than an instance of a class is known as a static method.
- a static method created by using the keyword **static** before the method name.
- The main advantage of a static method is that we can call it without creating an object.
- It can access static data members and also change the value of it.
- It is used to create an instance method.
- It is invoked by using the class name.
- The best example of a static method is the **main()** method.

Static Method - Example

```
public class Display
{
public static void main(String[] args)
{
show();
}
static void show()
{
System.out.println("It is an example of static method.");
}
}
```