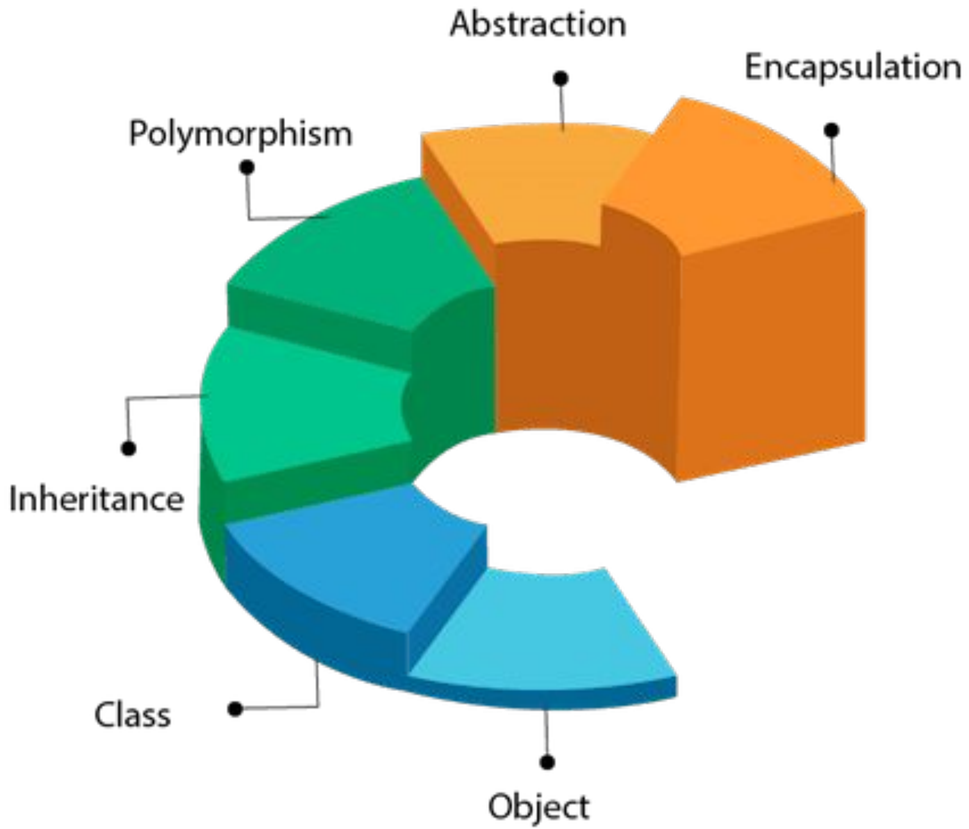
OOPs (Object-Oriented Programming System)



Classes & Objects

Can you name your favorite cricketer?

Segregate the nouns & verbs

MS DHONI

List them down

Mahendra Singh Dhoni

40 Years

Brown

Batsman

Indian

Batting

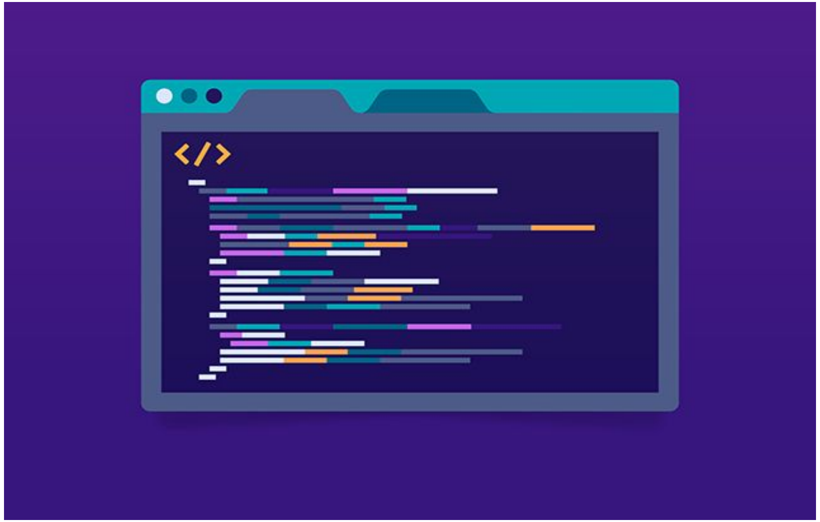
Wicket Keeping

Leading the team

Strategic decision making

Focus & determining

Let’s get technically started!



Class and object

What is Class?

- A class is a template or blueprint that is used to create objects.
- A class consists of data members and methods.

What is Object?

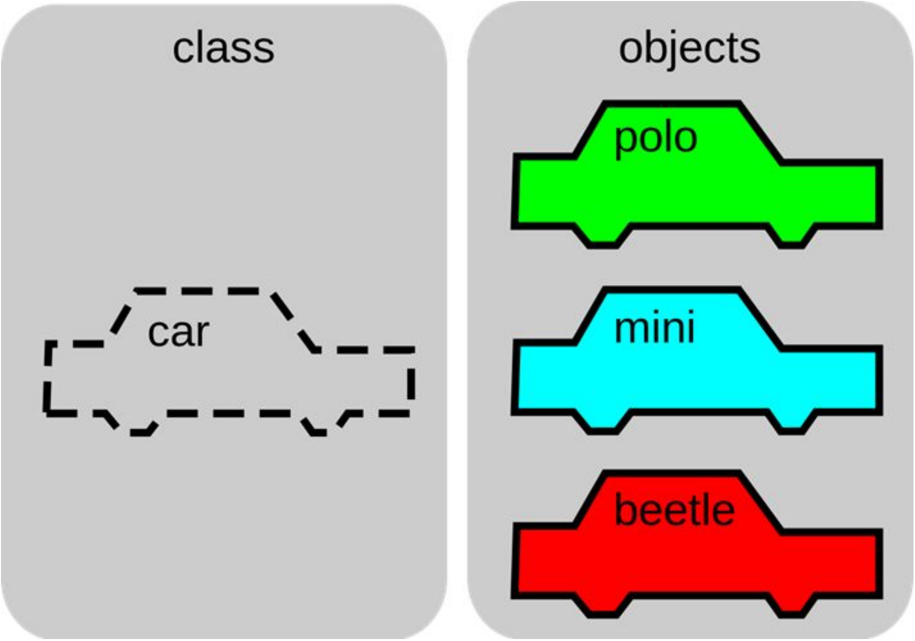
- Any real-world entity that has state and its behavior.

For Example:

A **car** is an object. It has states (name, color, model) and its behavior (changing gear, applying brakes).

Class & object

- A **blueprint** from which you can **create** an individual **object**
- **Any entity** that has:
  - **State**
  - **Behavior**
  - **Identity**
- **State --> data member**
- **Behavior --> method**
- **Identity --> unique Id(name)**



DEFINING A CLASS IN JAVA

Syntax: `

```
public class class_name
{
      Data Members;
      Methods;
}
```

Example:

```
public class Car
 {
   public:
   double color;
   double model;
 }
```

Cricketer

```
class Cricketer{
    //Instance variables
    String name = “Dhoni”;
    int age = 39;
    String color = “Brown”;
    String role = “Batsman”;
    String nationality = “Indian”;
```

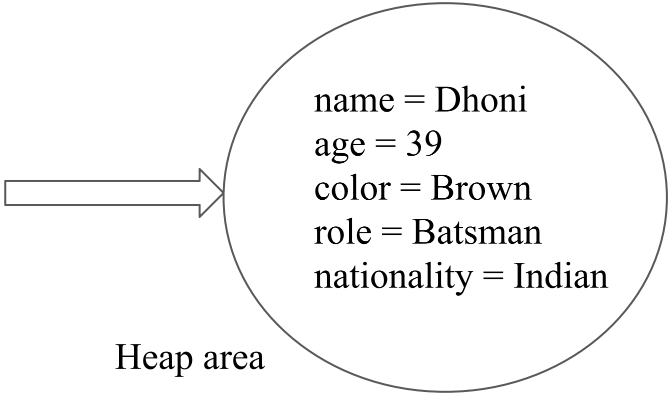
```
void Display(){
    System.out.println(“I LOVE
    Criket”);
    }//end of method
}//end of class
```

Class Instantiation

- Object is an instance of a **class**
- Process of creating an object from a **class** is called **instantiation**.
- Now that you have defined a new **class** type --> Cricketer, it is time to **instantiate** an object of that type.
- **new keyword** is used to initialize/create an object.
- **new** operator is followed by a call to a constructor, which initializes the **new** object.

Example:

```
Cricketer c = new Cricketer();
```



name = Dhoni
age = 39
color = Brown
role = Batsman
nationality = Indian

Heap area

Know the difference!

Example:01

```
public class Demo{
    int x;
}
class TestDemo{
public static void main(String[]
args) {
        Demo myObj = new Demo();
        myobj.x = 40;
        System.out.println(myObj.x);
    }
}
```

Example:02

```
 public class Demo {
    int x;
    public static void main(String[] args) {
        Demo myObj1 = new Demo();
        Demo myObj2 = new Demo();
        myobj1.x = 24;
        myobj2.x = 55;
        System.out.println(myObj1.x);
        System.out.println(myObj2.x);
    }
}
```

Cricketer - code snippet

```
class Cricketer{
     String name = “Dhoni”;
     int age = 39;
     String color = “Brown”;
     String role = “Batsman”;
     String nationality= “Indian”;
     void Display(){

    System.out.println(“I LOVE
    Criket”);
    }}//end of criketerDetails

    method
```

```
public static void main(String args[]){
    System.out.println(“main method
started”);
       Cricketer c = new Cricketer();
       c.Display ();
       System.out.println(“main method
ended”);
    }//end of main method
}//end of Cricketer class
```

3 WAYS TO INITIALIZE OBJECT

1. By reference variable
2. By method
3. By constructor

By reference variable

```
class Student {
    int  id;
    String  name;
}
```

```
class  Main {
    public  static  void  main(String  args[]) {
        Student  s1 = new  Student();
        s1.id = 101;
        s1.name = "John";
        System.out.println(s1.id + " " + s1.name);
    }
}
```

By method

```java
class Student {
    int  rollno;
    String  name;
    void  insertRecord(int  r,  String  n) {
        rollno = r;
        name = n;
    }
    void  displayInformation() {
        System.out.println(rollno + " " + name);
    }
}
```

By method - continued.,

```
class Main {
    public  static  void  main(String
args[]) {
        Student  s1 = new  Student();
        Student  s2 = new  Student();
        s1.insertRecord(111, "Karan");
        s2.insertRecord(222, "Aryan");
        s1.displayInformation();
        s2.displayInformation();
    }
}
```

LOGICAL SNIPPETS

```
Class Test {
 public static void main(String[] args) {
  Test obj = new Test();
  obj.start();
 }
 void start() {
  String stra = "do";
  String strb = method(stra);
  System.out.print(": " + stra + strb);
 }
 String method(String stra) {
  stra = stra + "good";
  System.out.print(stra);
  return " good";
 }
}
```

Knowledge check

Which of the following statement(s) is/are correct?

X: A class is like a blue print and we can create as many objects using that class.

Y: Every object should belong to a class, since we can not create objects with out a class.

A. X only

B. Y only

C. Both X and Y are correct

D. Both X and Y are incorrect

Answer: C

Knowledge check

Which of the following class declaration is correct?

```
/* X */
class Student
{
    String name;
    int marks;
    char section;
}
```

```
/* Y */
Class Student
{
    String name;
    int marks;
    char section;
}
```

A. X only

B. Y only

C. X and Y both are correct

D. X and Y both are incorrect

Answer: A

Knowledge check

Which of the following is a valid declaration of an object of class Student?

```
1. Student obj = new Student;
2. Student obj = new Student();
3. obj = new Student();
4. new Student obj;
```

A. 1 only

B. 2 only

C. 1 & 2

D. 3 & 4

Answer: B

Knowledge check

What will be the output of the following program?

```
class Apple {
     int weight;
}
class Main {
     public static void main(String args[]) {
          Apple a1 = new Apple();
          Apple a2 = new Apple();
          a1.weight = 1;
          a2.weight = 2;
          a2 = a1;
          System.out.println(a2.weight);
     }
}
```

A. 1
B. 2
C. Error
D. 1

Answer: A

Knowledge check

```
class Company {
    String name;
    public static void main(String[] args) {
        Company c = new Company();
        Company c1 = c;
        c1.name = “C2TC”;
        c = null;
        System.out. println(c.name);
    }
}
```

Knowledge check

```java
public class Company{
    String name;
    public static void main(String[] args) {
           Company c1 = new Company();
           c1.name = " C2TC ";
           System.out.println(c1.name);
           c1 = new Company();
           System.out.println(c1.name);
    }

}
```

Knowledge check

```
public class Company {
    String name;
        public static void main(String[] args) {
            Company c1 = new Company();
            c1.name = " C2TC ";
            Company c2 = c1;
            c1 = null;
            System.out.println(c2.name);
            new Company();
            Company c3;
            c3.name = " C2TC ";
            System.out.println(c3.name);}}

}
```