OBJECT ORIENTED PROGRAMMING USING JAVA

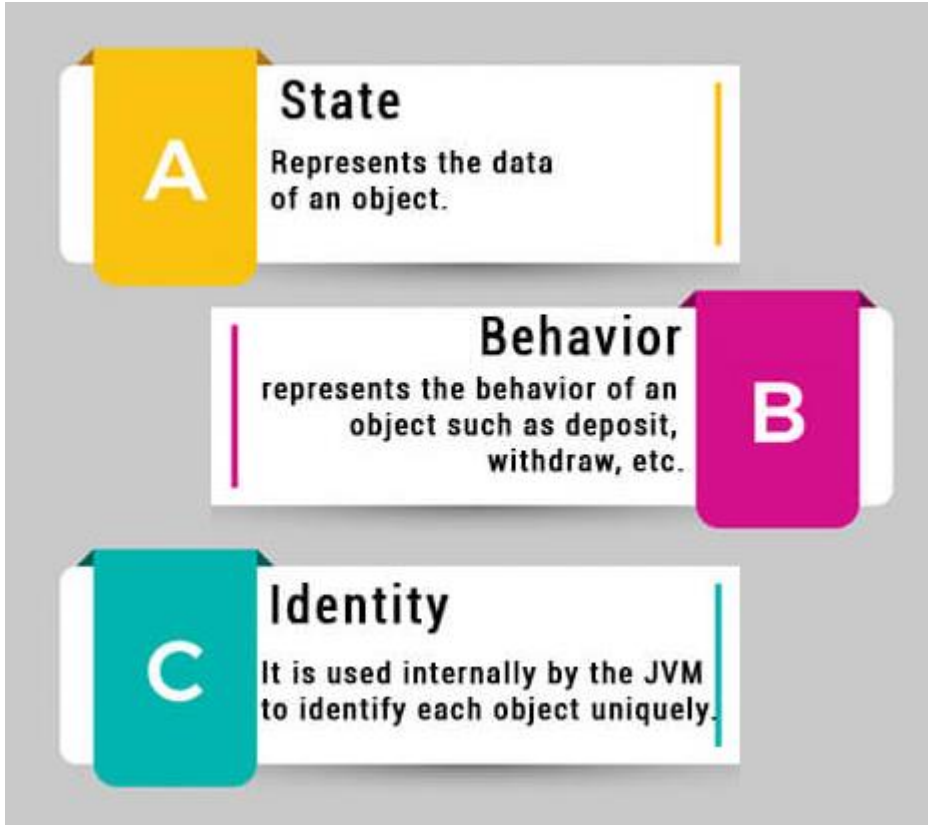
OUTLINE

- Objects & Classes
- Lifecycle of an Object

WHAT IS AN OBJECT IN JAVA?

- An entity that has state and behavior is known as an object
- e.g., chair, bike, marker, pen, table, car, etc.
- An object has three characteristics:
  1. State
  2. Behavior
  3. Identity

For Example, Pen is an object. Its name is Reynolds; color is white, known as its state. It is used to write, so writing is its behavior.



An object is an instance of a class.

CLASS

- A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.
- A class in Java can contain:
  - Fields
  - Methods
  - Constructors
  - Blocks
  - Nested class and interface

Syntax to declare a class:

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

OBJECT AND CLASS EXAMPLE:

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

OBJECT AND CLASS EXAMPLE:

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

Output:

0
null

LIFECYCLE OF AN OBJECT

- There are four parts in lifecycle of an Object:

  - ❖ Object creation
  - ❖ Object accessing
  - ❖ Object not accessing
  - ❖ Garbage collection

LIFECYCLE OF AN OBJECT (CONT..)

- **Object Creation:**
  - **We create an object by using new keyword.**
  - **We can initialize reference variable with the object.**

```
public class Creation
{
public static void main(string args[])
{
Creation abc=new Creation();
}
}
```

LIFECYCLE OF AN OBJECT (CONT..)

- **Object accessing:**
  - When we create an object , we can access it.
  - We can access it , till the statements/ conditions are true.
  - If were assign another object to an initialized reference variable, the previous object becomes inaccessible.

LIFECYCLE OF AN OBJECT (CONT..)

- **Object accessing (Example):**

```
class Object_Life
{
  int ID;
  String Name;
  void Record(int Identity, String Full_Name)
  {
    ID=Identity;
    Name=Full_Name;
  }
  void Display()
  {
    System.out.println("ID: "+ ID + ", " + "Name: " + Name);
  }
  public static void main(String args[])
  {
     Object_Life Emp=new Object_Life();
     Emp.Record(2019, "Suyel");
     Emp=new Object_Life();
     Emp.Record(2222, "Hello");
     Emp.Display();
  }
}
```

LIFECYCLE OF AN OBJECT (CONT..)

- **Object accessing (Example):**

```
class Object_Life
{
  int ID;
  String Name;
  void Record(int Identity, String Full_Name)
  {
    ID=Identity;
    Name=Full_Name;
  }
  void Display()
  {
    System.out.println("ID: "+ ID + ", " + "Name: " + Name);
  }
  public static void main(String args[])
  {
     Object_Life Emp=new Object_Life();
     Emp.Record(2019, "Suyel");
     Emp=new Object_Life();
     Emp.Record(2222, "Hello");
     Emp.Display();
  }
}
```

Output:

ID:2222, Name: Hello

LIFECYCLE OF AN OBJECT (CONT..)

- **Object not accessing:**
  - An object become inaccessible, if it goes out of scope.
  - Then, the object is no longer be referenced and compiler makes garbage collection for the object.

LIFECYCLE OF AN OBJECT (CONT..)

- **Garbage collection:**

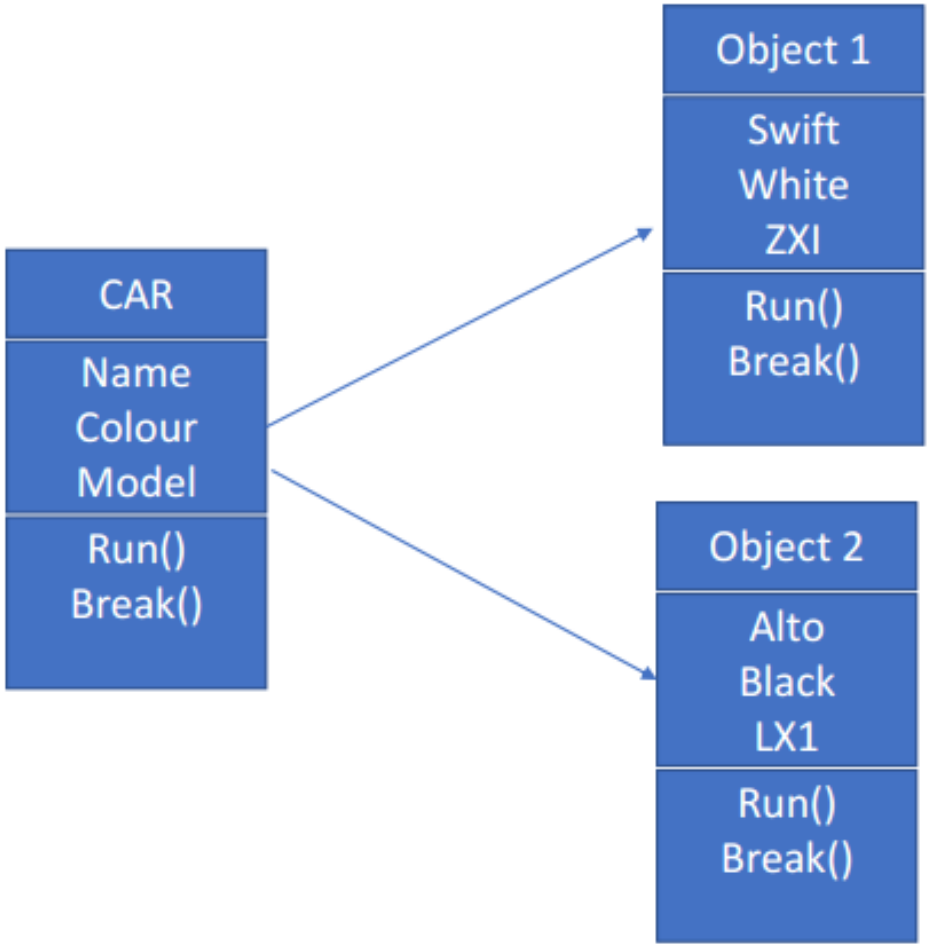
  - It is a process by which Java programs perform automatic memory management.
  - Java programs converts to bytecode that can be run on a JVM.
  - Objects are created on the heap(a portion of memory) dedicated to the program.
  - If any object is not needed in future, the garbage collector finds these unused objects and deletes them.

LIFECYCLE OF AN OBJECT (CONT..)

- **Garbage collection:**

  - **Unreachable objects**: An object has become unreachable , if it doesn't contain any reference to it.

```
Object_Life Emp=new Object_Life();
Emp.Record(2021,"BU");
Emp=null;
```


CAR
Name
Colour
Model
Run()
Break()
Object 1
Swift
White
ZXI
Run()
Break()
Object 2
Alto
Black
LX1
Run()
Break()

```
public class Car {
   String name;
   String model;
   String colour;

   void Run() {
   }

   void Break() {
   }
```

```
Car object1=new Car();
Car object2=new Car();
object1.name="Swift";
object1. model ="ZXI";
object1. colour ="White";
object1.Run();
object1.Break();


object2.name="Alto";
object2. model ="LXI";
object2. colour ="Black";
object2.Run();
object2.Break();
}
```

THANK YOU
?