SRI AKILANDESWARI WOMEN'S COLLEGE, WANDIWASH

CLASSES & OBJECTS

Class : II B.C.A

Mrs.R.SAIKUMARI
Assistant Professor
Department of Computer Applications

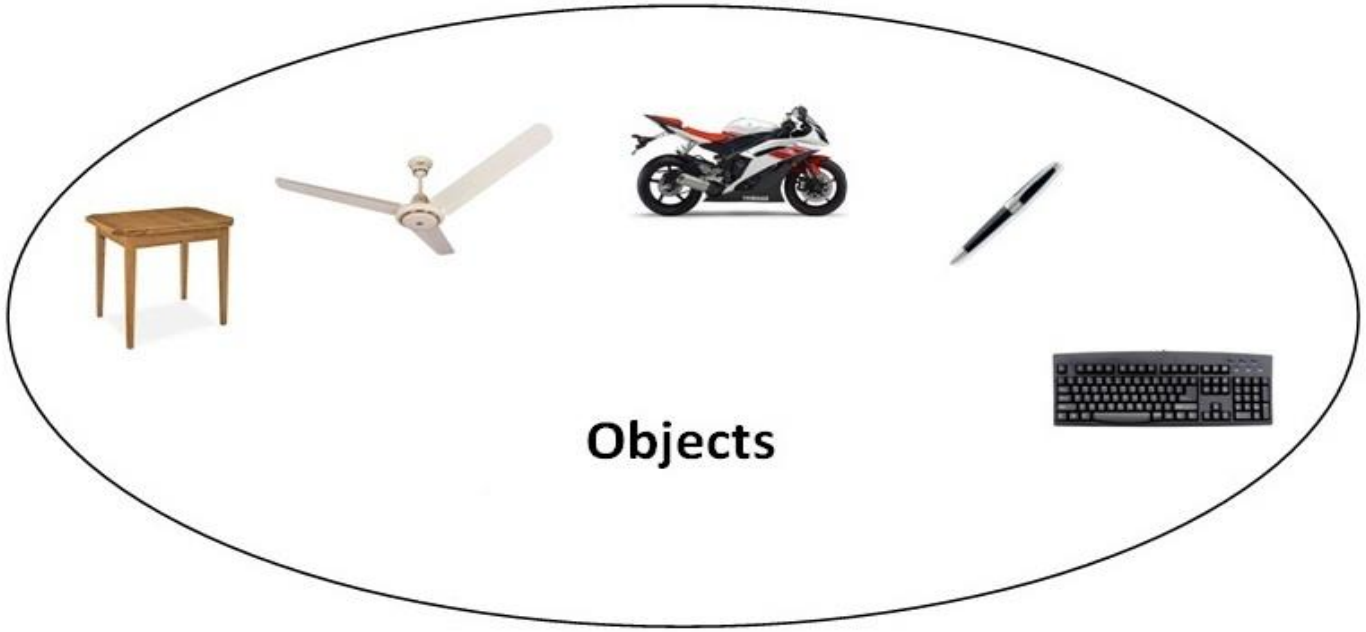
SWAMY ABEDHANADHA EDUCATIONAL TRUST, WANDIWASH

Object in Java

- An entity that has state and behavior is known as an object e.g. chair, bike, marker, pen, table, car etc. It can be physical or logical (tangible and intangible). The example of intangible object is banking system.

- An object has three characteristics:
- **State:** represents data (value) of an object.
- **Behavior:** represents the behavior (functionality) of an object such as deposit, withdraw etc.
- **Identity:** Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But, it is used internally by the JVM to identify each object uniquely.

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



- **Object is an instance of a class**. Class is a template or blueprint from which objects are created. So object is the instance(result) of a class.

- **Object Definitions:**
- Object is *a real world entity*.
- Object is *a run time entity*.
- Object is *an entity which has state and behavior*.
- Object is *an instance of a class*.

Class in Java

- 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;
}
```

New Keyword in Java

- The new keyword is used to allocate memory at run time. All objects get memory in Heap memory area.

Example

```
class Student
{
   int id;//field or data member or instance variable
   String name;
   public static void main(String args[])
        {
        //creating an object of Student
        Student s1=new Student( );
        //accessing member through reference variable
        System.out.println(s1.id);
        System.out.println(s1.name);
        }
}
```

Ways to Initialize Objects

- There are 3 ways to initialize objects in java.

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

Initialization through Reference

```
class Student
{   int id;
  String name;
 }
class TestStudent2
{   public static void main(String args[])
{    Student s1=new Student();
   s1.id=101;
  s1.name="Sonoo";
  System.out.println(s1.id+" "+s1.name);
//printing members with a white space
  }
 }
```

Initialization through Method

```
class Student{
 int rollno;
 String name;
 void insertRecord(int r, String n){
  rollno=r;
  name=n;
 }
 void displayInformation(){System.out.println(rollno+" "+name);}
}
class TestStudent4{
 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();
 }
}
```

Initialization through Constructor

```
class Employee{
    int id;
    String name;
    float salary;
    void insert(int i, String n, float s) {
        id=i;
        name=n;
        salary=s;
    }
    void display(){System.out.println(id+" "+name+" "+salary);}
}
public class TestEmployee {
public static void main(String[] args) {
    Employee e1=new Employee();
    Employee e2=new Employee();
    Employee e3=new Employee();
    e1.insert(101,"ajeet",45000);
    e2.insert(102,"irfan",25000);
    e3.insert(103,"nakul",55000);
    e1.display();
    e2.display();
    e3.display();
}
}
```