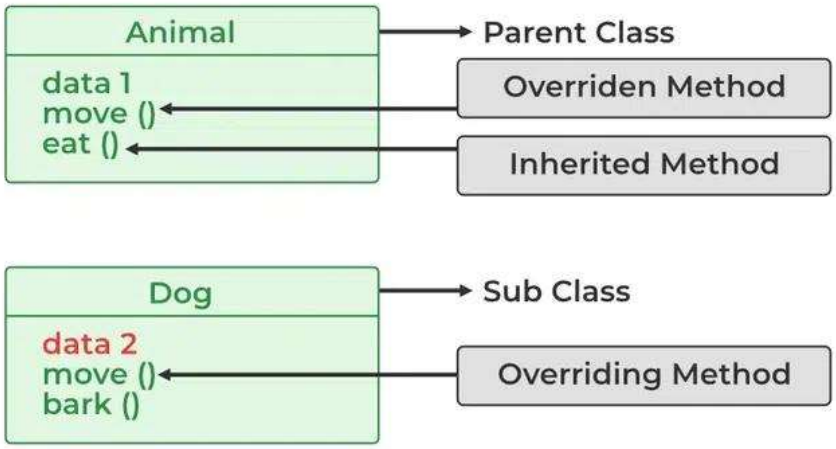
# Practical – 5 Inheritance -2

- ## Method Overriding

- Method overriding is used for runtime polymorphism.
- method in a subclass has the same name, the same parameters or signature, and the same return type(or sub-type) as a method in its super-class.
- it allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes




If an object of a parent class is used to invoke the method, then the version in the parent class will be executed, but if an object of the subclass is used to invoke the method, then the version in the child class will be executed.

```
// Base Class
class Parent {
  void show() { System.out.println("Parent's show()"); }
}
// Inherited class
class Child extends Parent {
  // This method overrides show() of Parent
  @Override void show()
  {      System.out.println("Child's show()");   }
}

// Driver class
class OverrideDemo {
  public static void main(String[] args)
```

```
  {
    // If a Parent type reference refer to a Parent object, then Parent' show is called
    Parent obj1 = new Parent();
    obj1.show();
    // If a Parent type reference refers  to a Child object Child's show()
    // is called. This is called RUN TIME  // POLYMORPHISM.
    Parent obj2 = new Child();
    obj2.show();
  }
}
```

## Constructor with inheritance

While calling constructor of the child class, parent class's default constructor is called.

```
class Parent  {

   private String name;
   Parent()
   { name=null; }
   Parent(String name)
   {   System.out.println("Parent Constructor");
       this.name=name;
   }
    } // end of parent


class Child extends Parent
{ String name;
  int age;
  Child() {
   age=0; name=null;
  }
  Child(String name){
   this.name = name;  }
}
 public class OverrideDemo1{
   public static void main(String[] arg){
```

```
        Child c1 = new Child("Child 1");
      c1.print();
   }
}
```

Output :

Parent Constructor Default constructor

Child class

Child 1

- **Super() key word**

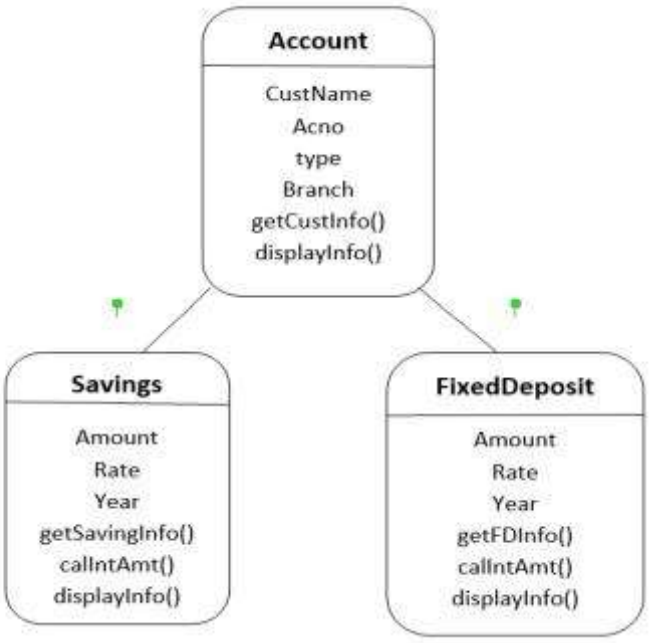
The **super keyword in Java** is a reference variable that is used to refer immediate parent class

```
class Person
{
 private    String name;
   private int age;
   Person(String name, int age){
      this.name = name;
      this.age= age;
   }
   void display()
    {
      System.out.println("Name :" + name);
      System.out.println("Age :" + age);
    }

}
class Employee extends Person
 {
    int salary;
    int exp;
```

```
   Employee(String name, int age, int salary, int exp){
     super(name,age);  // calling parent class constructor
     this.salary = salary;
     this.exp=exp;
   }
  void display(){
     super.display();  // call method of parent class
     System.out.println("Salary :" + salary);
     System.out.println("experience :" + exp);
      }
}

 public class SuperDemo{
  public static void main(String arg[]){
     Employee e1 = new Employee("AAAA",33,30000,2);
     e1.display();
  }
 }
```

Exercise :

| | |
|---|---|
| 1 | Write a java program for<br>Class Item<br>Property : itemname,price<br>Method . parameterize Constructor, default constructor, display()<br>Class Order inherits Item<br>Property : noofunit<br>Method : parameterize constructor(), getUnit() , cal_price();<br>Create object of Item class and object of Order class. And call methods. |
| 2. | Write a java program which demonstrate the working of Method overriding.<br>Class Shape<br>Method : area() -> display message of calling shape area method.<br>disp() : display message “inside the shape class”<br>Class Circle inherits shape<br>Property : radius<br>Method : area() return the area of circle. , disp() display the properties of circle<br>Class Rectangle inherits shape<br>Properties : length,width,height<br>Method : area( ) return the area of rectangle. disp() display the properties of rectangle.<br><br>Create a object of shape , circle and rectangle and call area() and disp() method of ever objects. |
| 3. | Write a java program for given situation . (by using super keyword)<br>ceate class Course :<br>Prporty , course name, noof semester<br>Methods : constructor disp_course() ,<br>Class Subject inherits Course.<br>Property : subname, subcode,maxcredit<br>Method : constructor, setter methods for property and getter method for property.<br>Getter methods return the appropriate value. , display_sub()<br>Class student inherits Subject<br>Property : name,marks, semester<br>Method : constructor , cal_percentage() , disp_stud() ,<br>Student() constructor takes the all the values of course,subect and students and pass it to appropriate constructor.<br>Create object of subject and display all details including course,subject and students, with the object of students. |
| 3 | W.A.P for the inheritance and override displayInfo() method and use super keyword in constructor to inherit parent class features . |




For savings account, Rate = 3.5 (fix)
Calculate interest = (amount* rate * year)/100
For Fixed Deposit,
If Years <=2 rate is 6.5
If year >2 and <=5 Rate is 7.5
If year >5 and <=10 Rate is 8.5 if Year is >10 Rate is 9