

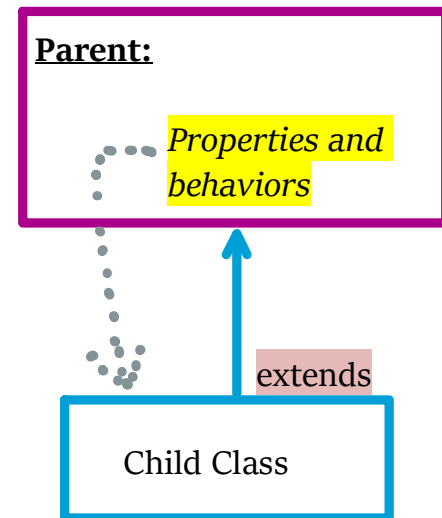
# Inheritance ( Is - A Relationship )

Is a Relationship is a concept in which one class can act as a *parent* and another can act as a *child*.

The child class will be able to access all the of the parent class directly.

This is done with the help of **extends** keyword.

Basically the process of a child class extends the properties and behaviors of a parent class is called as Inheritance.



```
class Parent // is a relationship
{
    public static String Pvar1 = "Parent Static Variable";
    public String Pvar2 = "Parent Instance Variable" ;

    static
    {
        System.out.println("Parent [ SIB ] - Static Initializer / Anonymous block");
    }

    {
        System.out.println("Parent [ IIB ] - Instance Initializer block");
    }

    public Parent()
    {
        System.out.println("Parent Object is Instantiated");
    }

    public static void Ptest1()
    {
        System.out.println("This is a parent static method");
    }

    public void Ptest2()
    {
        System.out.println("This is a parent instance method");
    }
}
```

```
class Friend // has a relationship
{
    public static String Fvar1 = "Friend Static Variable";
    public String Fvar2 = "Friend Instance Variable" ;

    static
    {
        System.out.println("Friend [ SIB ] - Static Initializer / Anonymous block");
    }

    {
        System.out.println("Friend [ IIB ] - Instance Initializer block");
    }
}
```

```

public Friend()
{
    System.out.println("Friend Object is Instantiated");
}

public static void Ftest1()
{
    System.out.println("This is a Friend static method");
}

public void Ftest2()
{
    System.out.println("This is an Friend instance method");
}
}

// ! the child is accessing the parent
class Child extends Parent { }

public class I1 //Driver class
{
    // Todo : Call all the members of a parent class using the child object

    public static void main(String[] args)
    {
        Child c= new Child();

        // ~ Anonymous blocks and constructors get executed implicitly

        System.out.println(Child.Pvar1+"\n"+c.Pvar2);
        Child.Ptest1();
        c.Ptest2();

        System.out.println("\n*\n**\n***\n****\n****\n***\n**\n*\n");

        Friend f = new Friend();

        // ? here we are using the borrowed data

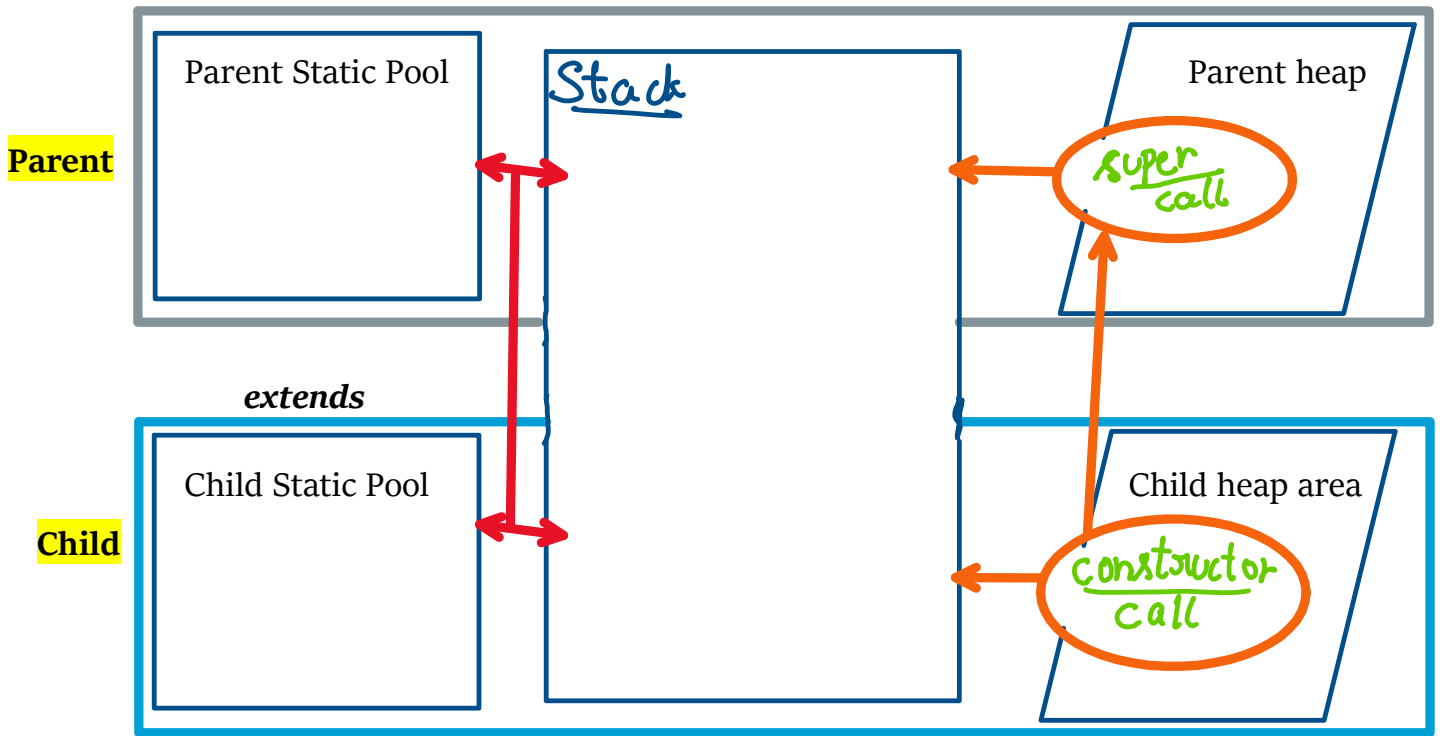
        System.out.println(Friend.Fvar1+"\n"+f.Fvar2);
        Friend.Ftest1();
        f.Ftest2();
    }
}

```

**How is inheritance performed..?**

**or**

**How does the *extends* keyword work..?**



```
class P1
```

```
{
```

```
    public P1()
```

```
    {
```

```
        System.out.println("Parent object is instantiated");
```

```
    }
```

```
}
```

```
public class C1 extends P1
```

```
{
```

```
    public C1()
```

```
    {
```

```
        //super() ;// super call statement  
        //calling the parent constructor
```

```
        System.out.println("Child object is instantiated");
```

```
    }
```

```
    public static void main(String[] args)
```

```
    {
```

```
        // creating the child instance
```

```
        new C1(); //calling the constructor of child
```

```
    }
```

Create a parent instance

Creates a child instance

The child will hold both the parent & child ref

```
new C1(); // calling the constructor of child
```

```
}  
}
```

The child will hold both the parent & child ref

## Example 2

```
class Google //Driven class  
{  
    public String username;  
    public String password;  
  
    public Google()  
    {  
        System.out.println("~Account is created~");  
    }  
    public Google(String username, String password)  
    {  
        this(); //this call statement , it calls the constructor of current class  
        this.username=username;  
        this.password=password;  
    }  
    public boolean login(String u, String p)  
    {  
        return ( u==username && p==password ) ;  
    }  
    public void search(String key)  
    {  
        System.out.println("Searching for "+key);  
  
        System.out.println("\n\nPlease click Here to View the results");  
    }  
}  
class Gmail extends Google  
{  
    public Gmail(String username, String password)  
    {  
        //super call statement  
        super(username, password);  
    }  
    public void Emailing()  
    {  
        System.out.println("Send or Receive emails");  
    }  
}  
public class Account //driver  
{  
    public static void main(String[] args)  
    {  
        Gmail ac1 = new Gmail("Qspiders", "Qsp@123");  
  
        // ! entered user details  
        String u="Qpiders", p="Qsp@123";  
        if( ac1.login(u, p) )  
        {  
            System.out.println("Login successfull");  
            ac1.Emailing();  
        }  
        else  
            System.out.println("Invalid Credentials");  
    }  
}
```

```
    }  
}
```

```
[Running] cd "c:\Programming\OOP\" && javac Account.java && java Account
```

```
~Account is created~  
Login successfull  
Send or Recive emails
```

```
[Done] exited with code=0 in 1.961 seconds
```

```
[Running] cd "c:\Programming\OOP\" && javac Account.java && java Account
```

```
~Account is created~  
Invalid Credentials
```

```
[Done] exited with code=0 in 2.076 seconds
```