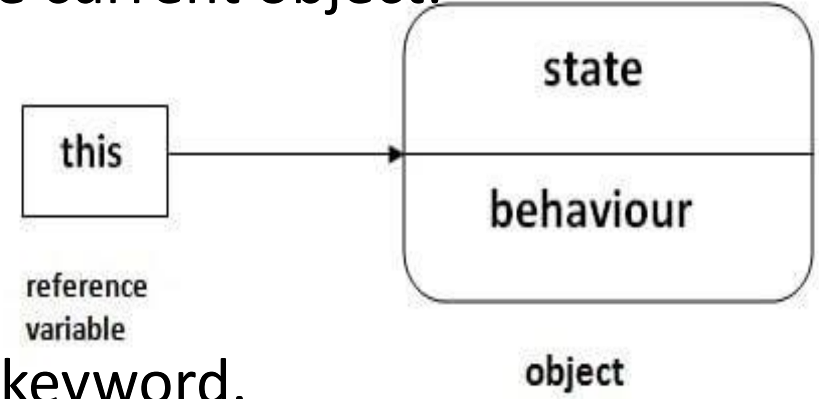
Inheritance in java

17-06-2020

Things we know

- **<u>This Keyword</u>**

  There can be a lot of usage of **java this keyword**. In java, this is a **reference variable** that refers to the current object.



Usage of java this keyword

- Here is given the 6 usage of java this keyword.
- this can be used to refer current class instance variable.
- this can be used to invoke current class method (implicitly)
- this() can be used to invoke current class constructor.
- this can be passed as an argument in the method call.
- this can be passed as argument in the constructor call.
- this can be used to return the current class instance from the method.

```
Class   abc
{
                    public abc()  //Default Constructor
                    {
                              System.out.print(“abc class constructor”);
                    }
                    public abc(int x)
                    {
                              System.out.print(“\nabc argument constructor”);
                    }
                    public void show()
                    {
                              return(this);  //this can be return
                    }
                    public static void main(String arg[])
                    {
                              abc   obj=new   abc();
                              obj.show();
                                        or
                              this.show(this);       //this can pass as argument
                              this();
                              this(5);
                    }
}
```

Final Keyword

The **final keyword** in java is used to restrict the user. The java final keyword can be used in many context. Final can be:

- variable
- method
- class

The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only. We will have detailed learning of these. Let's first learn the basics of final keyword.

1) Java final variable

If you make any variable as final, you cannot change the value of final variable(It will be constant).

- There is a final variable speedlimit, we are going to change the value of this variable, but It can't be changed because final variable once assigned a value can never be changed.

```
class Bike9
{
    final int speedlimit=90;//final variable
      void run()
      {
                speedlimit=400;  //error
      }
      public static void main(String args[])
      {
                Bike9 obj=new  Bike9();
                obj.run();
      }
}//end of class
```

Java static keyword

- The **static keyword** in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes. The static keyword belongs to the class than an instance of the class.

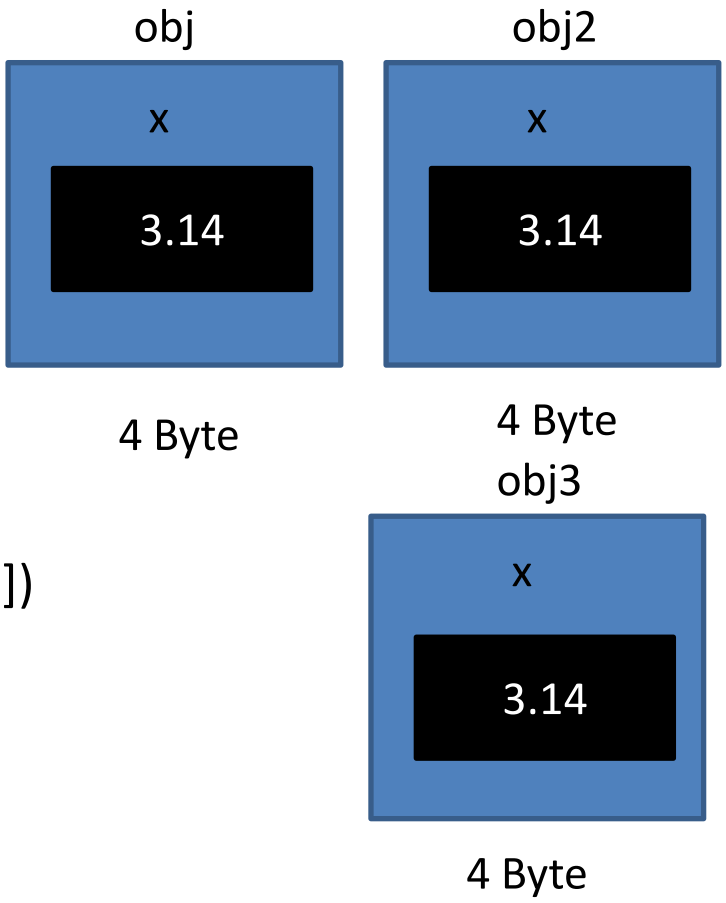
The static can be:

- Variable (also known as a class variable)
- Method (also known as a class method)
- Block
- Nested class

  The static variable can be used to refer to the common property of all objects (which is not unique for each object), for example, the company name of employees, college name of students, etc.

- The static variable gets memory only once in the class area at the time of class loading.
- Advantages of static variable
- It makes your program **memory efficient** (i.e., it saves memory).

```
Class abc
{
        int pi;
        public abc()
        {
                pi=3.14f;
        }
}
Class ex
{
        public static void main(string arg[])
        {
                abc  obj=new abc();
                abc  obj1=new abc();
                abc obj3=new abc();
        }
}
```



obj

x

3.14

4 Byte

obj2

x

3.14

4 Byte

obj3

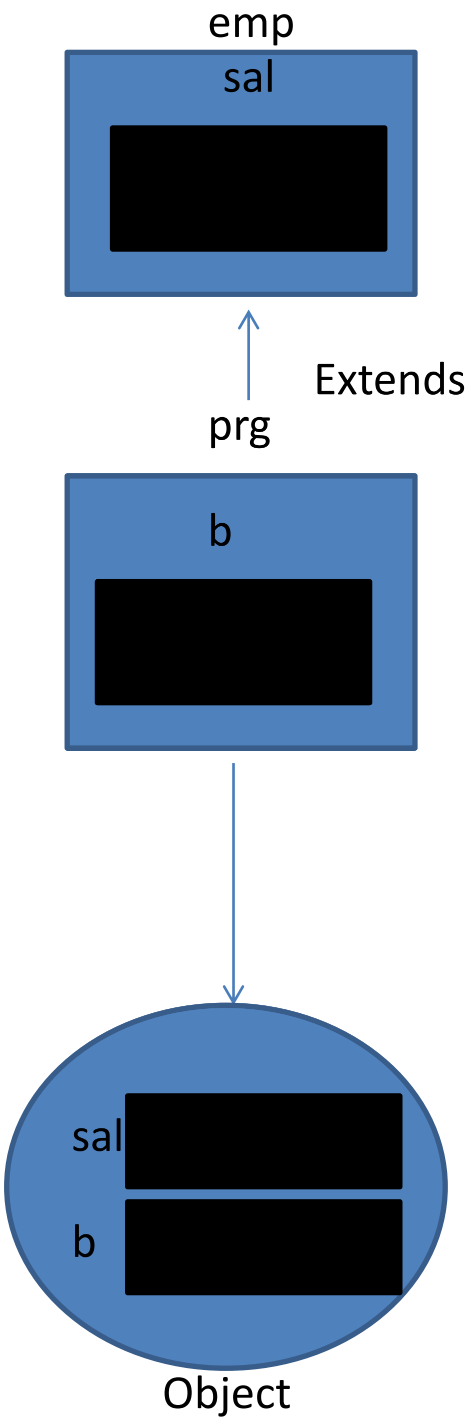
x

3.14

4 Byte

Inheritance in java

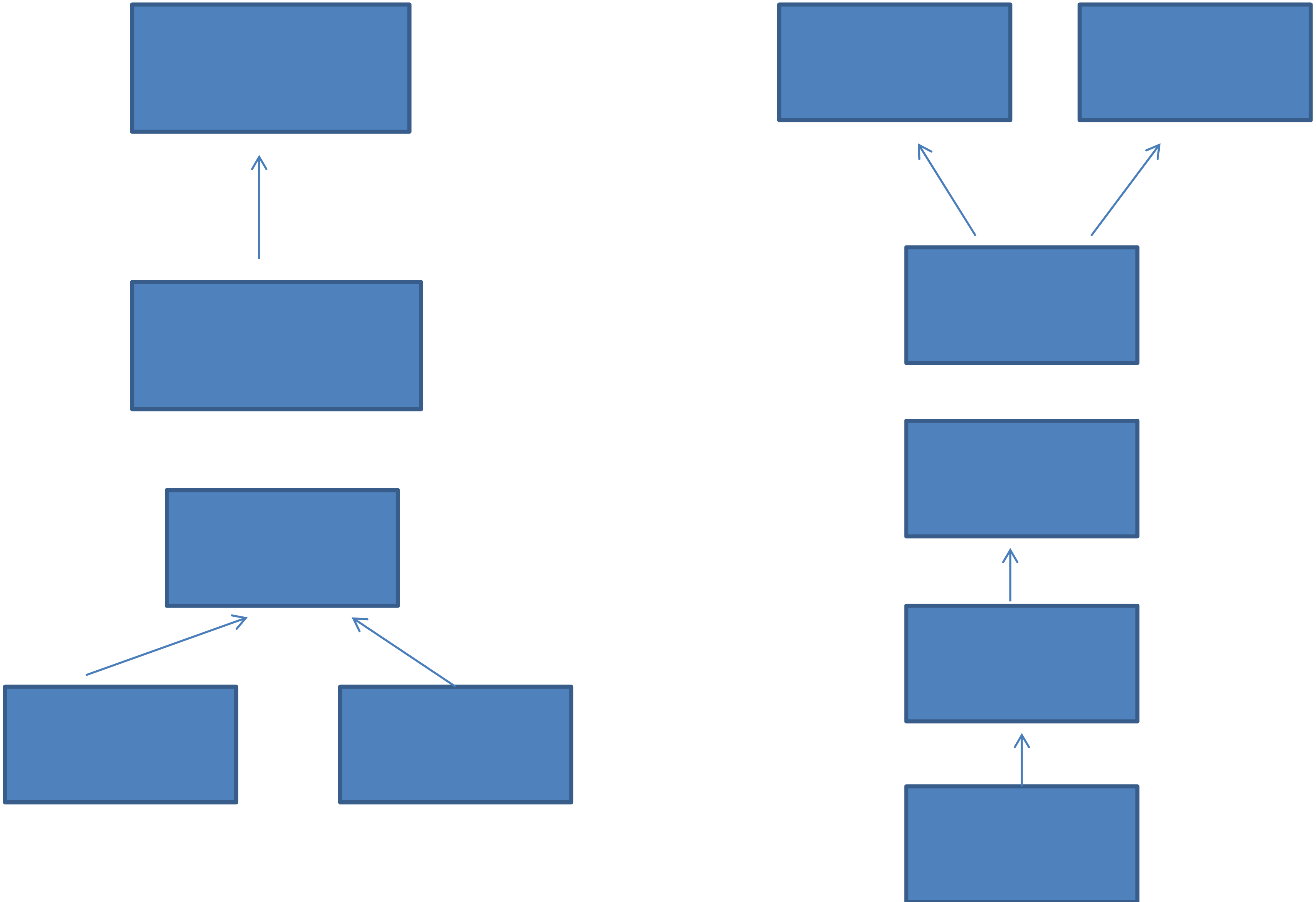
- Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.
- The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).
- **Inheritance in Java** is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).


emp
sal
[redacted]
Extends
prg
b
[redacted]
sal
[redacted]
b
[redacted]
Object

- Types of inheritance: (OOPS)

1) Single Inheritance
2) Multiple Inheritance
3) Multi Level Inheritance
4) Hierarchy Inheritance
5) Hybrid Inheritance
6) Multipath Inheritance

Inheritance



How to create Inheritance

```
Class  base
{
        data member;
        member function;
}
Class derived  extends   base
{
        data member;
        member function;
}
```

Note:

1. extends keyword use with only one class.
2. Extends keyword use with derived class.
3. Using extends we can implements only single inheritance.

- The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.Inheritance represents the **IS-A relationship** which is also known as a *parent-child* relationship.
- **Why use inheritance in java**
  - **For Method Overriding (so runtime polymorphism can be achieved).**
  - **For Code Reusability.**

Terms used in Inheritance

- **Class:** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- **Sub Class/Child Class:** Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- **Super Class/Parent Class:** Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- **Reusability:** As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.
- **The syntax of Java Inheritance**

```
class Subclass-name extends Superclass-name
{
   //methods and fields
}
```

The **extends keyword** indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality. In the terminology of Java, a class which is inherited is called a parent or superclass, and the new class is called child or subclass.

Java Inheritance Example

As displayed in the above figure, Programmer is the subclass and Employee is the superclass. The relationship between the two classes is **Programmer IS-A Employee**. It means that Programmer is a type of Employee.

```
class Employee
{
         float salary=40000;
}
class Programmer extends Employee
{
         int bonus=10000;
        public static void main(String args[])
        {
                    Programmer p=new Programmer();
                    System.out.println("Programmer salary is:"+p.salary);
                    System.out.println("Bonus of Programmer is:"+p.bonus);
        }
}
```

O/P:

```
Programmer salary is:40000.0
Bonus of programmer is:10000
```

The super keyword

- The **super** keyword is similar to **this** keyword. Following are the scenarios where the super keyword is used.
- It is used to **differentiate the members** of superclass from the members of subclass, if they have same names.
- It is used to **invoke the superclass** constructor from subclass.

Q) Write a java code accept personal information like name, add, age for personal class also create student class and accept roll, course, sem. Display student information with name, add, course, roll, sem,age.