FAQ's on Inheritance

1. What is inheritance in Java?

Acquiring the features from existing entities is known as "inheritance". Existing entity is called as super or parent class and the new entity that is acquiring the features of an existing entity is called as sub-class or child class.

The process of inheriting the properties and behaviours from the one object to another object is known as java.

Creating new class from existing class using **IS-A** relationship is known as "inheritance".

2. How do you implement inheritance practically in Java?

There are two keywords in java to implement is-a relationship

- Extends and
- Implements.

3. What is the purpose of inheritance?

Inheritance offers code reusability and theirby Rapid Application Development (RAD) is possible.

4. What are the generalized and specialized classes in Java?

The top level or super classes are known as "generalized class". It contains common data and common behaviour.

The low-level or sub-level classes are known as "specialized classes". It contains more specific data

```
Ex:
class Person{
     int name;
     int age;
 }
Class Student extends Person{
        int rollNo;
        int marks;
 }
class Employe extends Person{
      int empId;
     float empSal;
```

```
    }
```

In this class hierarchy **Person** is generalized class and **Student** and **Employe** are the specialized classes.

5. What are the types of inheritance?

The following are the types of inheritances:

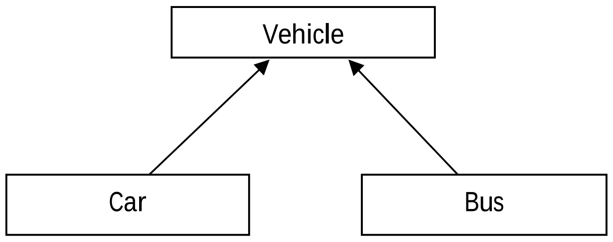
- Single inheritance, and
- Multiple inheritances.

Single inheritance: If a class is having only one parent class that is known as "single inheritance". The following are two special cases of single inheritance supported by Java:

- Hierarchical Inheritance and
- Multi-level Inheritance.

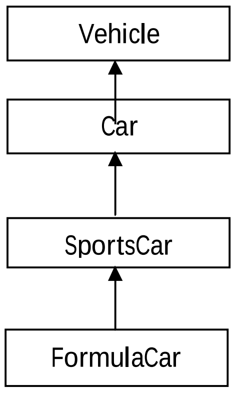
Hierarchical inheritance: If a class has more than one sub-class that form of single inheritance specially known as "hierarchical inheritance".

Example:



Multi-level inheritance: If a sub-class acts as super-class for another sub-class such single inheritance is known as "multi-level inheritance".

Example:



Code Example of single inheritance:

```
        Package org.talentsprint.inheritance;
        Class A {
          int x:
```

```
  int y;
  int get(int p, int q){
  x=p; y=q; return(0);
  }
  void Show(){
  System.out.println(x);
  }
}
class B extends A{
  public static void main(String args[]){
  A a = new A();
  a.get(5,6);
  a.Show();
  }
  void display(){
  System.out.println("B");
  }
}
```

Multiple Inheritance:

The mechanism of inheriting the features of more than one base class into a single class is known as multiple inheritances. Java does not support multiple inheritances but the multiple inheritances can be achieved by using the interfaces by implementing more than one interface in a class.

```
Package org.talentsprint.inheritance;
class A {
  int x;
  int y;
  int giveDetails(int p, int q){
         x=p;
         y=q;
         return(0);
  }
  void show(){
  System.out.println(x);
  System.out.println(y);
  }
}
class B extends A{
  void view(){
  System.out.println("Hai from class B");
  }
}
class C extends B{
  void display(){
  System.out.println("Hai from class C");
  }
public static void main(String args[]){
             C c = new C();
             c.giveDetails(5,6);
             c.show();
             c.view();
             c.display();
     }
```

```
            }
```

6. What is the purpose of 'super' keyword in Java?

The following are the major uses of super keyword they are:

- If your method overrides one of its super class's methods, you can invoke the overridden method through the use of the keyword super.
- Using super keyword we can explicitly call the immediate super class constructor from the sub-class constructor.
- From sub-class, to access an instance variables of super class where sub-class is also having a variable with the same name.

7. What are the differences between 'this' and 'super' keyword?

Using 'this' from within one class constructor, a call can be made to other constructor of the same class. Whereas, using 'super' keyword super class constructor is called from sub-class constructor.

8. What are the rules to be followed while overriding a method?

The following are the rules to be followed while overriding a method:

- The overriding method has the same name, number and type of parameters, and return type as the method it overrides.
- For the child method accessibility should not be reduced than that of parent class method, equal is ok or higher is ok.
- In the child method exception specification extra checked exception classes should not be listed than that of parent class method, equal is ok or less is ok.

9. What is the base class of all classes?

The *java.lang.Object* is the base class of all classes.

10. Does Java support multiple inheritances?

Java doesn't support multiple inheritances.

11. How to define a constant variable in Java?

The variable should be declared as static and final. So only one copy of the variable exists for all instances of the class and the value can't be changed also. `static final int PI = 2.14;` is an example for constant.

12. What is the purpose of declaring a variable as ‘final’?

A final variable's value can't be changed. The ‘final’ variables should be initialized before using them.

13. What is the impact of declaring a method as final?

A method declared as final can't be overridden. A sub-class can't have the same method signature with a different implementation.

14. I don't want my class to be inherited by any other class. What should I do?

You should declare your class as final. But you can't define your class as final, if it is an abstract class. A class declared as final can't be extended by any other class.

15. Can you give few examples of final classes defined in Java API?

`java.lang.String`, `java.lang.Math` are the ‘final’ classes.

16. How is final different from finally and finalize()?

The ‘final’ is a modifier which can be applied to a class or a method or a variable. ‘final’ class can't be inherited, final method can't be overridden and final variable can't be changed.

The ‘finally’ is an exception handling code section which gets executed whether an exception is raised or not by the try block code segment.

The ‘finalize()’ is a method of Object class which will be executed by the JVM just before garbage collecting object to give a final chance for resource releasing activity.

17. Does a class inherit the constructors of its superclass?

A class does not inherit constructors from any of its superclasses.

18. What is Overriding?

When a class defines a method using the same name, return type, and arguments as a method in its superclass, the method in the class overrides the method in the superclass.

When the method is invoked for an object of the class, it is the new definition of the method that is called, and not the method definition from superclass. Methods may be overridden to be more public, not more private.

19. How are this() and super() used with constructors?

The ‘this()’ method is used to invoke a constructor of the same class. Whereas the ’super()’ is used to invoke a superclass constructor.

20. What modifiers are allowed for methods in an Interface?

Only public and abstract modifiers are allowed for methods in interfaces.

21. What are some alternatives to inheritance?

Delegation is an alternative to inheritance.

Delegation means that you include an instance of another class as an instance variable, and forward messages to the instance. It is often safer than inheritance because it forces you to think about each message you forward, because the instance is of a known class, rather than a new class, and because it doesn't force you to accept all the methods of the super class: you can provide only the methods that really make sense. On the other hand, it makes you write more code, and it is harder to re-use (because it is not a subclass).

22. Does a class inherit the constructors of its superclass?

A class does not inherit constructors from any of its superclasses.

23. What restrictions are placed on method overloading?

Two methods may not have the same name and argument list but different return types.

24. What is method overloading & method overriding?

Method overloading: When a method in a class having the same method name with different arguments is said to be method overloading. Method overriding: When a method in a class having the same method name with same arguments is said to be method overriding.

25. What is diff. between overloading & overriding?

- In overloading, there is a relationship between methods available in the same class whereas in overriding, there is relationship between a superclass method and subclass method.
- Overloading does not block inheritance from the superclass whereas overriding blocks inheritance from the superclass.
- In overloading, separate methods share the same name whereas in overriding, subclass method replaces the superclass.
- Overloading must have different method signatures whereas overriding must have same signature.

26. What is meant by Inheritance and what are its advantages?

Inheritance is the process of inheriting all the features from a class. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.

27. What is the difference between superclass and subclass?

A super class is a class that is inherited whereas sub class is a class that does the inheriting.

28. What modifiers may be used with top-level class?

The public, abstract and final can be used for top-level class.