Commonly Asked Java Programming Interview Questions | Set 2

Frequently Asked Java Interview Questions | Set 1

Can we Overload or Override static methods in java ?

- **Overriding :** Overriding is related to run-time polymorphism. A subclass (or derived class) provides a specific implementation of a method in superclass (or base class) at runtime.
- **Overloading:** Overloading is related to compile time (or static) polymorphism. This feature allows different methods to have same name, but different signatures, especially number of input parameters and type of input paramaters.
- **Can we overload static methods?** The answer is **'Yes'**. We can have two ore more static methods with same name, but differences in input parameters
- **Can we Override static methods in java?** We can declare static methods with same signature in subclass, but it is not considered overriding as there won't be any run-time polymorphism. Hence the answer is **'No'**. Static methods cannot be overridden because method overriding only occurs in the context of dynamic (i.e. runtime) lookup of methods. Static methods (by their name) are looked up statically (i.e. at compile-time).

Read more

Why the main method is static in java?

The method is static because otherwise there would be ambiguity: which constructor should be called? Especially if your class looks like this:

```
public class JavaClass
{
  protected JavaClass(int x)
  {   }
```

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

     }
}
```

Should the JVM call new JavaClass(int)? What should it pass for x? If not, should the JVM instantiate JavaClass without running any constructor method? because that will special-case your entire class – sometimes you have an instance that hasn't been initialized, and you have to check for it in every method that could be called. There are just too many edge cases and ambiguities for it to make sense for the JVM to have to instantiate a class before the entry point is called. That's why main is static.

What happens if you remove static modifier from the main method?
Program compiles successfully . But at runtime throws an error "NoSuchMethodError".

What is the scope of variables in Java in following cases?

- **Member Variables** (Class Level Scope) : The member variables must be declared inside class (outside any function). They can be directly accessed anywhere in class
- **Local Variables** (Method Level Scope) : Variables declared inside a method have method level scope and can't be accessed outside the method.
- **Loop Variables** (Block Scope) : A variable declared inside pair of brackets "{" and "}" in a method has scope withing the brackets only.

Read more

What is "this" keyword in java?
Within an instance method or a constructor, this is a reference to the current object — the object whose method or constructor is being called. You can refer to any member of the current object from within an instance method or a constructor by using this.
Usage of this keyword

- Used to refer current class instance variable.
- To invoke current class constructor.
- It can be passed as an argument in the method call.
- It can be passed as argument in the constructor call.
- Used to return the current class instance.
- Used to invoke current class method (implicitly)

What is an abstract class? How abstract classes are similar or different in Java from C++?
Abstract classes are classes that contain one or more abstract methods. An abstract method is a method that is declared, but contains no implementation. Abstract classes may not be instantiated, and require subclasses to provide implementations for the abstract methods.

- Like C++, in Java, an instance of an abstract class cannot be created, we can have references of abstract class type though.
- Like C++, an abstract class can contain constructors in Java. And a constructor of abstract class is called when an instance of a inherited class is created
- In Java, we can have an abstract class without any abstract method. This allows us to create classes that cannot be instantiated, but can only be inherited.
- Abstract classes can also have final methods (methods that cannot be overridden). For example, the following program compiles and runs fine.

Read more

Which class is the superclass for every class ?
Object class

Can we overload main() method?
The main method in Java is no extra-terrestrial method. Apart from the fact that main() is just like any other method & can be overloaded in a similar manner, JVM always looks for the method signature to launch the program.

- The normal main method acts as an entry point for the JVM to start the execution of program.
- We can overload the main method in Java. But the program doesn't execute the overloaded main method when we run your program, we need to call the overloaded main method from the actual main method only.

Read more

What is object cloning?
Object cloning means to create an exact copy of the original object. If a class needs to support cloning, it must implement java.lang.Cloneable interface and override clone() method from Object class. Syntax of the clone() method is :

```
protected Object clone() throws CloneNotSupportedException
```

If the object's class doesn't implement Cloneable interface then it throws an exception 'CloneNotSupportedException' .

Read more

How is inheritance in C++ different from Java?

1. In Java, all classes inherit from the Object class directly or indirectly. Therefore, there is always a single inheritance tree of classes in Java, and Object class is root of the tree.
2. In Java, members of the grandparent class are not directly accessible. See this G-Fact for more details.
3. The meaning of protected member access specifier is somewhat different in Java. In Java, protected members of a class "A" are accessible in other class "B" of same package, even if B doesn't inherit from A (they both have to be in the same package).
4. Java uses *extends* keyword for inheritance. Unlike C++, Java doesn't provide an inheritance specifier like public, protected or private. Therefore, we cannot change the protection level of members of base class in Java, if some data member is public or protected in base class then it remains public or protected in derived class. Like C++, private members of base class are not accessible in derived class.
   Unlike C++, in Java, we don't have to remember those rules of inheritance which are combination of base class access specifier and inheritance specifier.
5. In Java, methods are virtual by default. In C++, we explicitly use virtual keyword. See this G-Fact for more details.
6. Java uses a separate keyword *interface* for interfaces, and *abstract* keyword for abstract classes and abstract functions.
7. Unlike C++, Java doesn't support multiple inheritance. A class cannot inherit from more than one class. A class can implement multiple interfaces though.
8. In C++, default constructor of parent class is automatically called, but if we want to call parametrized constructor of a parent class, we must use Initializer list. Like C++, default constructor of the parent class is automatically called in Java, but if we want to call parameterized constructor then we must use super to call the parent constructor.

See examples here

Why method overloading is not possible by changing the return type in java?

In C++ and Java, functions can not be overloaded if they differ only in the return type . The return type of functions is not a part of the mangled name which is generated by the compiler for uniquely identifying each function. The No of arguments, Type of arguments & Sequence of arguments are

the parameters which are used to generate the unique mangled name for each function. It is on the basis of these unique mangled names that compiler can understand which function to call even if the names are same(overloading).

Can we override private methods in Java?

No, a private method cannot be overridden since it is not visible from any other class. Read more

What is blank final variable?

A final variable in Java can be assigned a value only once, we can assign a value either in declaration or later.

```
final int i = 10;
i = 30; // Error because i is final.
```

A **blank final** variable in Java is a final variable that is not initialized during declaration. Below is a simple example of blank final.

```
// A simple blank final example
final int i;
i = 30;
```

Read more

What is "super" keyword in java?

The super keyword in java is a reference variable that is used to refer parent class objects. The keyword "super" came into the picture with the concept of Inheritance. Whenever you create the instance of subclass, an instance of parent class is created implicitly i.e. referred by super reference variable.

Various scenarios of using java super Keyword:

- super is used to refer immediate parent instance variable
- super is used to call parent class method
- super() is used to call immediate parent constructor

Read more

What is static variable in Java?

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

The static can be:

- variable (also known as class variable)
- method (also known as class method)
- block
- nested class

Differences between HashMap and HashTable in Java.

1. HashMap is non synchronized. It is not-thread safe and can't be shared between many threads without proper synchronization code whereas Hashtable is synchronized. It is thread-safe and can be shared with many threads.
2. HashMap allows one null key and multiple null values whereas Hashtable doesn't allow any null key or value.
3. HashMap is generally preferred over HashTable if thread synchronization is not needed

Read more

How are Java objects stored in memory?

In Java, all objects are dynamically allocated on **Heap**. This is different from C++ where objects can be allocated memory either on Stack or on Heap. In C++, when we allocate abject using new(), the object is allocated on Heap, otherwise on Stack if not global or static.

In Java, when we only declare a variable of a class type, only a reference is created (memory is not allocated for the object). To allocate memory to an object, we must use new(). So the object is always allocated memory on heap. Read more

What are C++ features missing in Java?

Try to answer this on your own before seeing the answer – here.

See also

- Java Multiple Choice Questions
- Practice Coding Questions
- Java articles

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice