

FAQ's on Classes and Objects

1. What is a class in Java?

In Java, a class:

- Is a user-defined data type.
- Is a blue print /plan or template for a proposed object.
- Contains both variables and methods.
- Defines structure and behaviour of the objects that are going to be created from it.
- Is a keyword.
- Is similar to 'structure' of C language.
- Represents collection of objects.

2. What is an object in Java?

In Java, an object:

- Is an instance of a class.
- Is a fundamental unit of data representation in an object-oriented system.
- Holds data and the code acts upon the data.

3. What are the properties of an object?

An object has chiefly three properties. They are:

- **State:** Data members of an object and their current values put together is nothing but state of the object. i.e. data of the object is known as state of the object.
- **Behaviour:** The functional part of an object is nothing but object behaviour. i.e. methods of an object are object behaviour.
- **Identity:** The name of the object with which it is uniquely identified is known as "identity of an object".

4. How do you create a class?

The syntax to create a class is:

```
class classname{
    accessibility mode datatype variblename1; //default
    accessibility mode is: default in Java
    accessibility mode datatype variblename2;
    accessibility mode datatype variblename3;
    .....
    .....
    accessibility mode return type methodname1(argument list){
        // body
    }
}
```

```

    }
    accessibility mode return type methodname2(argument list){
        // body
    }
    accessibility mode return type methodname3(argument list){
        // body
    }
}

```

Example:

```

class FirstProgram{
    Int number;
    Byte num;
    Char ch;
    Public void display(){
        System.out.println(number);
        System.out.println(num);
        System.out.println(ch);
    }
}

```

5. How to create object in Java?

We can create object of a class using the following syntax:

```
ClassName reference name = new ConstructorName();
```

Where:

- **ClassName** : it is the name of the class for which object is to be created
- **reference name**: it is class type variable pointing to the object of class.
- **new** : It is a dynamic memory allocation operator in Java used to allocate the memory for the class members and methods
- **ConstructorName**: It is a special method having the same the name as class.

Example: Object creation for the above class:

```
FirstProgram firstreference = new FirstProgram();
```

Note: In Java all statements must be enclosed in a class.

6. How to access class data members and methods in Java?

In order to access the class members, we have to create a new class (not mandatory but advisable). In the new class, define public static void main() method and create object of that class.

Using the dot '.' (The member accessing operator) we can access both data and methods of an Object or a Class

For example:

```
class ClassDemo{
    Int number;
    byte num;
    char ch;
    Public void display(){
        System.out.println(number);
        System.out.println(num);
        System.out.println(ch);
    }
}
class RunProgram{
    Public static void main(String args[]) {
        ClassDemo firstreference = new ClassDemo ();
        firstreference.display();
    }
}
```

7. What are the steps involved in the object creation in Java?

The steps involved in the object creation process are:

Step -1: A new operator constructs the Java class object.

Step -2: JVM provides the default values to the instance variables of the object.

Step -3: JVM calls a special method called constructor on the object.

Step -4: A new operator binds the object to the reference variable, i.e. new operator returns the object address to be stored in a specialized pointer known as "reference".

8. What are the instance variables in Java?

The instance variables in Java are the non-static data members of a class. They represent the structure of the object and contribute to the size of the object.

If object does not exist then the instance variables do not exist physically.

For each instance of a class one separate copy of the variables are created and hence we call them as instance variables.

9. What are the instance methods in Java?

The non-static methods in a class are known as "instance methods". These methods can be called on the instances (objects) and hence the name. For each instance of a class one separate copy of instance method is not created.

10. What is default value in Java?

Default values are the values provided by the JVM when no value is supplied to the instance variables of class, by default JVM provides the initial values to them.

11. What are the default values provided by the JVM to the instance variables?

JVM provides the following default values to all primitive type variables:

- All integral types (int, short, long, byte) is : zero (0)
- All decimal types (float, double) is :0.0
- For char type it is null character : \u0000 (where u stands for Unicode)
- For Boolean type it is: false

12. What are local variables?

The local variables are the variables which are declared in the method of class. They should be initialised before accessing them. The scope of these variables is confined to their respective methods.

13. Does JVM provide any default values to the local variables?

No. JVM does not provide any default values to local variables.

14. How JVM does gives default values for the user-defined variables?

For class scoped user-defined variables, JVM provides null as default value.

15. How can you classify variables based on the data types?

Variables are of two types based on its type:

- Primitive type
- User-defined type (user-defined types are not objects in Java unlike C++).

Example:

```
class VariableTypes{
    int num; //primitive instance variable
    Integer number; //user-defined instance variable
    void method(){
        Float real; //user-defined local variable
        int b; //primitive local variable
        String name; //user-defined local variable}}

```

16. Can a class be declared as static?

No. Classes cannot be defined as static. Only, methods, variables and blocks of code can be declared as static.

17. When will you define a method as static?

A method is defined as static when it needs to be accessed even before the creation of the object of the class.

18. What are the restrictions imposed on a static method or a static block of code?

A static method should not refer to instance variables without creating an instance, and cannot use "this" operator to refer the instance.

19. I want to print "Hello" even before main() is executed. How will you achieve that?

Print the statement inside a static block of code. Static blocks get executed when the class gets loaded into the memory and even before the creation of an object. Hence it will be executed before the main() method. And it will be executed only once.

20. What is the importance of static variable?

The static variables are the class level variables where all the objects of the class refer to the same variable. If one object changes the value then the change gets reflected in all the objects.

21. Can we declare a static variable inside a method?

The static variables are class level variables and they can't be declared inside a method. If declared, the class will not be compiled.

22. What are the local, member and class variables?

The variables declared within a method are called "local" variables. The variables declared within the class, i.e. not within any methods are called "member" variables (or global variables). Whereas, the variables declared within the class, i.e. not within any methods and are defined as "static" are called class variables.

23. What is the difference between a static and a non-static inner class?

A non-static inner class may have object instances that are associated with the instances of the class's outer class. A static inner class does not have any object instances.

24. What is an object's lock and which objects have locks?

An object's lock is a mechanism that is used by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the object's lock. All objects and classes have locks. A class's lock is acquired on the class's Class object.

25. Which non-Unicode characters may be used as the first character of an identifier?

'\$' and '_' may appear as the first character of an identifier.

26. Can an anonymous class be declared as implementing an interface and extending a class?

An anonymous class may implement an interface or extend a superclass, but it may not be declared to do both.

27. What modifiers can be used with a local inner class?

'final' or 'abstract' can be used with a local inner class.

28. When does the compiler supply a default constructor for a class?

If no other constructors are provided, compiler supplies a default constructor for the class.

29. What are the legal operands of the instanceof operator?

The left operand is an object reference or null value and the right operand is a class, interface, or array type.

30. Are values 'true' and 'false' can be classified as Java keywords?

No.

31. What happens when you add a double value to a String?

When you add a double value to a string, it results into a string object.

32. What is the difference between inner class and nested class?

When a class is defined within a scope of another class then it becomes an inner class. If the access modifier of the inner class is static then it becomes a nested class.

33. What is the difference between a constructor and a method?

A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself. It has no return type, and it is invoked using the new operator.

A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.

34. What is the purpose of garbage collection in Java and when is it used?

The purpose of garbage collection is to identify and discard objects that are no longer needed by a program, so that their resources can be reclaimed and reused.

A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used.

35. What is static in Java?

Static means one per class, not one for each object no matter how many instance of a class might exist. This means that you can use them without creating an instance of a class. The static methods are implicitly final because overriding is done based on the type of the object and static methods are attached to a class, not to an object.

A static method in a superclass can be shadowed by another static method in a subclass as long as the original method was not declared final. However, you can't override a static method with a non-static method. In other words, you can't change a static method into an instance method in a subclass.

36. What if I do not provide the String array as the argument to the method?

Program gets compiled, but at the same time it also throws a runtime error: "NoSuchMethodError".

37. What is the first argument of the String array in main() method?

The String array is empty. It does not have any element. This is unlike C/C++ where the first element by default is the program name.

38. If I do not provide any arguments on the command line, then the String array of main() method will be empty or null?

It is empty. But not null.

39. How many objects are created in the following piece of code?

```
MyClass c1, c2, c3;
```

```
c1 = new MyClass ();
```

```
c3 = new MyClass ();
```

Only 2 objects are created, c1 and c3 because the reference c2 is only declared but not initialized.

40. Can a public class MyClass be defined in a source file named YourClass.java?

No. The source file name, if it contains a public class, must be the same as the public class name itself with a '.java' extension.

41. What are wrapped classes?

The Wrapped classes are classes that allow primitive types to be accessed as objects.

42. Does garbage collection guarantee that a program will not run out of memory?

Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection.

43. Can an object's finalize() method be invoked by the garbage collector while it is reachable?

An object's finalize() method cannot be invoked by the garbage collector while the object is still reachable. However, an object's finalize() method may be invoked by other objects.

44. For which value a variable of the String type gets initialized automatically?

The default value of a String type is null.

45. Can a Byte object be cast to a double value?

No. A Byte object cannot be cast to a primitive value.

46. If a class is declared without any access modifiers, where may the class be accessed?

A class that is declared without any access modifiers is said to have package access. This means that the class can only be accessed by other classes and interfaces that are defined within the same package.

47. What is the SimpleTimeZone class?

The SimpleTimeZone class provides support for the Gregorian calendar.

48. What modifiers can be used with a local inner class?

A local inner class may be final or abstract.

49. What is the difference between static and non-static variables?

A static variable is associated with the class as a whole rather than with specific instances of a class.

Non-static variables take on unique values with each object instance.

50. What classes of exceptions may be thrown by a throw statement?

A throw statement may throw any expression that may be assigned to the throwable type.

51. What are the Object and Class classes used for?

The Object class is the highest-level class in the Java class hierarchy. The Class class is used to represent the classes and interfaces that are loaded by a Java program.

52. Can an unreachable object become reachable again?

An unreachable object may become reachable again. This can happen when the object's `finalize()` method is invoked and the object performs an operation which causes it to become accessible to reachable objects.

53. When is an object subject to garbage collection?

An object is subject to garbage collection when it becomes unreachable to the program in which it is used.

54. What are Class, Constructor and Primitive data types?

Class is a template for multiple objects with similar features and it is a blue print for objects. It defines a type of object according to the data the object can hold and the operations the object can perform. Constructor is a special kind of method that determines how an object is initialized when created. Primitive data types are 8 types and they are: byte, short, int, long, float, double, boolean, char.

55. What is an Object and how do you allocate memory to it?

Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.

56. What is the difference between constructor and method?

Constructor will be automatically invoked when an object is created whereas method has to be called explicitly.

57. What are the methods and how are they defined?

Methods are functions that operate on instances of classes in which they are defined. Objects can communicate with each other using methods and can call methods in other classes. Method definition has four parts. They are name of the

method; type of object or primitive type the method returns, a list of parameters and the body of the method. A method's signature is a combination of the first three parts mentioned above.

58. What is Garbage Collection and how to call it explicitly?

When an object is no longer referred to by any variable, Java automatically reclaims memory used by that object. This is known as garbage collection. The `System.gc()` method may be used to call it explicitly.

59. What is `finalize()` method?

The `finalize()` method is used just before an object is destroyed and can be called just prior to garbage collection.

60. What are inner class and anonymous class?

Inner Class: The classes defined in other classes, including those defined in methods are called inner classes. An inner class can have any accessibility including private.

Anonymous Class: An anonymous class is a class defined inside a method without a name and is instantiated and declared in the same place and cannot have explicit constructors.

61. Can you have an inner class inside a method and what variables can you access?

Yes, we can have an inner class inside a method. We can access final variables.