# **Static Variable, Methods and Classes**

**Can abstract class have static variable in it ?**

Yes, an abstract class can have static variables in it.

**When will you define a method as static?**

When a method needs to be accessed even before the creation of the object of the class then we should declare the method as static.

**What is the restriction 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.

**non-static method cannot be referenced from a static context?**

public class Test

{

/\*\* Non Static main method with String[] args\*\*/

public static void main(String[] args)

{

welcome();

}

void welcome()

{

System.out.println("Welcom to JavaInterviewPoint");

}

}

The welcome() method which we tried calling is an instance-level method, we do not have an instance to call it . static methods belong to the class, non-static methods belong to instances of the class and hence it throws the error ”non-static method cannot be referenced from a static context“.

**Can a class be declared as static?**

We can not declare top level class as static, but only inner class can be declared static.

public class Test

{

static class InnerClass

{

public static void InnerMethod()

{ System.out.println("Static Inner Class!"); }

}

public static void main(String args[])

{

Test.InnerClass.InnerMethod();

}

}

//output: Static Inner Class!

**I want to print "Hello" even before main() is executed. How will you acheive 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.

**package** intquestions;

**class** testclass

{

**static** {

System.***out***.println("hello");

}

**public** **static** **void** main(String[] args) {

}

}

This will print the result as “hello”.

**What is the importance of static variable?**

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

**Can we declare a static variable inside a method?**

Static varaibles are class level variables and they can't be declared inside a method. If declared, the class will not compile.

**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 instances of the class's outer class. A static inner class does not have any object instances.

**What happens to a static variable that is defined within a method of a class ?**

Can't do it. You'll get a compilation error.

**How many static initializers can you have?**

As many as you want, but the static initializers and class variable initializers are executed in textual order and may not refer to class variables declared in the class whose declarations appear textually after the use, even though these class variables are in scope.

**Why main() method is public, static and void in java ?**

public : “public” is an access specifier which can be used outside the class. When main method is declared public it means it can be used outside class.

static : To call a method we require object. Sometimes it may be required to call a method without the help of object. Then we declare that method as static. JVM calls the main() method without creating object by declaring keyword static.

void : void return type is used when a method does’nt return any value . main() method does’nt return any value, so main() is declared as void.

**Explain the scope or life time of class variables or static variables?**

Static variables do not belong to instances of the class. We can access static fields even before instantiating the class. Static variable remain in memory till the life time of application.

**Explain about static imports in java?**

From Java 5.0 we can import static variables in to source file. Importing static member to source file is referred as static import. The advantage of static import is we can access static variables without class or interface name.

Syntax :

import static packagename.classname.staticvariablename;

Ex : import static com.abc.Employee.eno;

To import all static variables from a class in to our source file we use \*.

import static com.abc.Employee.\*