

Static Keyword**1) What are the different types of static elements that can exist in a class?**

Static variables, static blocks and static methods.

2) When should we declare a variable as static?

When a common copy of the value is required for all the objects of the class.

3) When should we create static blocks?

To initialise static variables and to execute a set of statements even before the execution of main() method.

4) When should we declare a method as static?

When the method is a utility method or generic method. When the method in real life is callable without any object.

5) When should we declare a variable as non static?

When we require a separate values for each variables of an object, then we should declare non static variables.

6) Why should we create non static blocks?

To initialize non static data and it also enables to count the number of objects of a class.

7) When should we declare a method as non static?

If it is not a utility method or if it is a specific method or if it is such a method which has to be accessed through a reference.

8) Can static method access non static data? Why ?

No. Because memory would not have been allocated for non static variables, no values would be present inside them.

9) Can static block access non static data? Why ?

No. Because memory would not have been allocated for non static variables, no values would be present inside them.

- 10) **Can static method access static data? Why?**
Yes
- 11) **Can static block access static data? Why?**
Yes
- 12) **Can non static method access non static data? Why ?**
Yes
- 13) **Can non static block access non static data? Why ?**
Yes
- 14) **Can non static method access static data? Why?**
Yes
- 15) **Can non static block access static data? Why?**
Yes
- 16) **Who initializes static data to default value?**
JVM.
- 17) **Who initializes non static data?**
JVM
- 18) **When does the static block get executed?**
Soon after the memory allocation of static variables , before the execution of main() method.
- 19) **When does the non static block get executed?**
After new operator creates an object and before calling the constructor, non static block gets executed.

20) From where would the static methods be loaded into the stack?

From the static segment.

21) From where would the non static methods be loaded into the stack?

Code segment

22) What is a class loader?

It is the part of JVM which loads all the static elements from code segment onto the static segment. The roles of class loader are,

- i) Loads all the static variables on static segment.
- ii) Loads static block onto the segment and executes it.
- iii) Loads static methods from code segment onto the static segment. However, class loader would not execute static methods. Rather static methods would be executed when the control enters into the method

23) Why should the main() be declared static?

If the main() method is not declared as static then class loader cannot load the main() method into the static segment. JVM after receiving the control from the class loader would automatically searches for the main() method in the static segment. If it is not present in static segment, then JVM cannot handover the control to the main() method and load it onto the stack segment. In other words the application never gets launched.

24) List practical applications of static keyword?

- i. Aadhar card-nationality
- ii. for students-university
- iii. In boys' school-gender
- iv. Institute-fees

25) I want to print "Hello" even before main() is executed. How can I achieve that?

If that message is present within the static block, then it would get executed even before the main() method.

Example:

```
class Demo
{
    public static void main (String [] args)
    {
        System.out.println("Inside main");
    }

    static
    {
        System.out.println("Hello");
    }
}
```

Output

Hello

Inside main

26) Can we have a static variable and a non static variable with the same name in a given class?

No

27) Can a class be declared as static?

Outer class cannot be declared as static. However, inner class (nested class) can be declared as static.

28) Can we use this keyword on a static variable?

No

29) Can we use this keyword on a non static variable?

Yes

30) Can we declare a static variable inside a static method?

No, if static variables declared inside the method, then it would become local variable and memory would be allocated only when the control

enters into the method and memory would be deallocated once the control leaves the method. IF the memory for static variable is deallocated in this manner then all the objects of the class cannot share the value of it. Memory for static variable has to be deallocated only after program terminates its execution.

31) Can we declare a static variable inside a non-static method?

No

32) What is the difference between class variables, instance variables and local variables?

For class variables, memory would be allocated only once. in other words only one copy of class variables would be created. All the objects of the class can access the same copy.

For instance variables, memory would be allocated separately for each object i.e. one copy per object.

Local variables would not be initialized automatically.

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

It cannot be created. Error would occur.

Example:

```
class Launch
{
    public static void main(String[] args)
    {
        static int a=10; // error
        System.out.println(a);
    }
}
```

34) How many static initializers can you have and in what order do they get executed?

We can have any number of static initializers (static blocks). they would get executed in the same order in which they are created .

35) Can main() method be overloaded?

Yes, provided the signature is different if they are present in the class. however, if they are present in different classes we can have multiple main methods with the same signature.

(Eg. Refer class notes)

36) What is the main role of static keyword?

It helps in Memory management

37) Can we apply static keyword on variables?

Yes

38) Can we apply static keyword on methods?

Yes

39) Can we apply static keyword on blocks?

Yes

40) Can we apply static keyword on class?

No, however, for inner class (nested class) static keyword can be apply.

41) Can we apply static keyword on nested class?

Yes

42) Can we apply static keyword on package?

Yes

43) Can we apply static keyword on constructor?

No

44) What is a static variable also known as?

Class variable

45) What is a static method also known as?

Class method or utility method or Generic method.

46) How can we count the number of objects that are created of a class?

(refer class notes)

47) Can we call a static method without using an object?

Yes, by using class name.

48) Can we call a static method using an object?

Yes

49) Can we call a non static method without using an object?

No

50) Can we call a non static method using an object?

Yes

51) Can we execute a program without main()?

NO

52) What is the difference between static variable and non static variable?

Static variable	Non static variable
Also called as class variable	Also called as instance variable
Memory is allocated in the static segment	Memory is allocated on heap segment
Memory would be deallocated only when the program terminates its execution	Memory would be deallocated when the object does not have any reference refereeing to it

A single copy would be created and all the objects have to share the single copy	A copy is created for each object of the class.
“this” keyword should not be used	“this” keyword can be used

53) What would happen if a top level class in java is declared as static?

Error would occur.

54) When are static variables initialized

Once class loader loads static variables from load code segment to static segment, static variables would get initialized.

55) When are non static variables initialized?

Once objects are created, during call to constructor, non static variables would be initialized.

56) Can static methods be overloaded?

Yes

57) What is the problem when static keyword is used in multithreaded environment?

Race condition occurs.

58) What are the disadvantages with static methods?

Static methods can not be overridden. They can not be used effectively in multithreaded environment.

59) Which design pattern can be implemented through static method?

Factory design pattern.

60) Name a few static classes in java?

Wrapper classes, java.lang.Math, java.lang.Collections etc.

61) Which keyword must be used along with static for creating constants?

“final” keyword.

62) What is the use of nested static class?

Packaging convenience.

63) What are the restrictions imposed on static methods?

Static methods cannot access non static data and cannot use this.

64) How can “one per class” policy enforced in java?

Using “static” keyword.

65) What is the difference between nested classes and inner classes?

If inner class is declared as static then it becomes nested.

66) Can the nested classes have access to the other members of the enclosing class?

No, but inner classes can access.

67) Can the inner class have access to the other members of the enclosing class?

Yes

68) What are the advantages of nested classes?

Increases encapsulation, makes packaging more streamlined, makes code readable and maintainable.