***Interview FAQs***

***new operator vs newinstance() method***

we can use new operator to create an object if we know class name at the beginning

EG:

Test t = new Test();

Student s = new Student();

Customer c = new Customer();

newinstance is a method present in class “Class”.

We can use newinstance method to create object if we don’t know class name at the beginning but dynamically at runtime.

EG:

**public** **class** Rough

{

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

{

Object o = Class.*forName*(args[0]).newInstance();

System.***out***.print("object created for"+o.getClass().getName());

}

}

i/p 1: Class Rough Student

o/p 1: object created for Student

i/p 2: Class Rough Customer

o/p 2: object created for Customer

i/p 3: Class Rough java.lang.String

o/p 3: object created for java.lang.String

***newInstance() method is deprived / removed from java now***

In the case of new operator based on our requirement we can invoke any constructor.

EG:

Test t = new Test();

Test t1 = new Test(10);

Test t2 = new Test(“Durga”);

But newInstance() internally calls no argument constructor. Hence to use newInstance method compulsory corresponding class should contain no argument constructor otherwise we will get runtime exception saying InstantiationException.

While using new operator at runtime if the corresponding .class file is not available then we will get runtime exception saying NoClassDefFoundError : Test

EG: Test t = new Test();

At runtime if Test.class file is not available then we will get runtime exception saying NoClassDefFoundError : Test

While using newInstance() at runtime if the corresponding .class file si not available then we will get runtime exception saying ClassNotFoundException : Test

EG: Object o = Class.forName(args[0]).newInstance();

i/p: java Test Test123

At runtime If Test123.class file is not available then we will get runtime exception saying ClassNotFoundException : Test123

|  |  |
| --- | --- |
| new | newInstance() |
| It is an operator in java | It is a method present in java.lang.Class |
| We can use new operator to create object if we know class name at the beginning | We can use this method to create object if we don’t know class name at the beginning and it is available dynamically at runtime |
| To use new operator class is not required to contain no-arg constructor | To use noInstance() compulsorily class should contain no-arg constructor otherwise we will get RE:InstantiationException |
| At runtime if class file is not available then we get RE: NoClassDefFoundError | At runtime if class file is not available then we get RE: ClassNotFoundException |

***ClassNotFoundException vs NoClassDefFoundException***

* For hardcoded class names , at runtime if the corresponding .class file is not available then we will get runtime exception saying NoClassDefFoundError which is unchecked.
* At runtime if Test.class file is not available then we will get runtime exception saying NoClassDefFoundError: Test

EG:

Test t = new Test();

* For dynamically provided class names at runtime if the corresponding .class file is not available then we will get runtime exception saying ClassNotFoundException which is checked exception.

EG:

Object o = Class.forName(args[0]).newInstance

i/p: Java Test Student

* At runtime If Student.class file is not available thenwe will get runtime exception saying ClassNotFoundException:Student

***isInstance() vs instanceof***

* instanceof is an operator in java
* We can use instanceof to check whether the given object is of particular type or not and we know the type at the beginning .

EG:

Thread t = new Thread();

SOP(t instanceof Runnable) 🡪True

SOP(t instanceof String) 🡪False

* Isinstance() is a method present in java.lang.Class.
* We can use isinstance() method to check whether the given object is of particular type or not and we don’t know the type at the beginning and it is available dynamically at runtime.
* EG:

Class : **public** **class** Rough

{

**public** **static** **void** main(String[] args) **throws** exception

{

Thread t = **new** Thread();

Object o = Class.*forName*(args[0]).~~newInstance~~();

System.***out***.print("object created for"+o.getClass().getName());

}

}

i/p: Java Test Runnable

o/p: true

i/p: Java Test java.lang.String

o/p: false

isInstance() is a method equivalent of instanceof operator.