1. What is difference between ClassNotFoundException and NoClassDefFoundError?

Both are Runtime Exception.

When using **n**ew **o**perator we will get **No**ClassDefFoundError

When using newInstance() method we will get ClassNotFoundException

2. Equality operator:

What are the conditions you have to apply equality operator?

If we apply equality operator, then their should be some relation between argument types either child to parent or parent to child or same type otherwise we will get compile time error saying in-comparable types.

Thread t1 = new Thread();

String str = new String();

Object obj = new Object();

System.out.println(t1 == obj);

System.out.println(str == obj);

System.out.println(t1 == str); //would result in compile-time error

Same goes for instanceof operator.

3. What is the evalution order of operands in Java?

There is no evalution order for operands in Java, there is only operator precedence and it will be executed from left to right.

4. Is it possible to apply increment and decrement operator to constanst like ++10;

Nope, we can apply incre/decrement operator only for variables but not for constants.

5. Is it possible to nest the incre/decrement operator like ++(++x)?

Nope it is not possible to nest incre/decrement operator.

6. Is it possible to apply incre/decrment operator to final variable?

Nope it is not possible.

7. Is it possible to apply incre/decrement operator to boolean variables?

Nope, except boolean variable we can apply incre/decrement operator to every primitive types.

8. What is the difference b/w b++ and b = b+1?

byte a = 10;

a = a + 1; //Incompatiable type, Required byte, found int

But in case of incre/decrement internal type casting will be performed.

a++; ==> (type of a ) (a+1)

9. Difference b/w instanceof and isInstance

instanceof -> we know the type in advance

isInstance() -> we don’t know the type in advance and it is a method equivalent of instanceof

Class.forName(args[0]).isInstance(t)

10. Difference b/w new and newInstance

new -> we know the type in advance

newInstance -> we don’t know the type in advance. This method need no-argument constructor.

Class.forName(args[0]).newInstance();