1)What is null pointer Exception?

answer:-NullPointerException is a RuntimeException. In Java, a special null value can be assigned to an object reference.

NullPointerException is thrown when an application attempts to use an object reference that has the null value. These include:

Calling an instance method on the object referred by a null reference.

Accessing or modifying an instance field of the object referred by a null reference.

If the reference type is an array type, taking the length of a null reference.

If the reference type is an array type, accessing or modifying the slots of a null reference.

If the reference type is a subtype of Throwable, throwing a null reference.

2)When “Unable to add window to Activity…. Is your Activity running”

error occurs?

anser-)My Activity is trying to create an AlertDialog which requires a Context as a parameter. This works as expected if I use:

AlertDialog.Builder builder = new AlertDialog.Builder(this);

However, I am leery of using "this" as a context due to the potential for memory leaks when Activity is destroyed and recreated

even during something simple like a screen rotation.

There are two easy ways to avoid context-related memory leaks.

The most obvious one is to avoid escaping the context outside of its own scope.

The example above showed the case of a static reference but inner classes and their implicit

reference to the outer class can be equally dangerous. The second solution is to use the Application context.

This context will live as long as your application is alive and does not depend on the activities life cycle.

If you plan on keeping long-lived objects that need a context, remember the application object.

You can obtain it easily by calling Context.getApplicationContext() or Activity.getApplication().

3)What is ArrayIndexOutOfBound Exception?

answer-)Assume, the marks of 50 students are stored in an array. A particular student marks are known by taking the roll number

(it becomes index number) from the keyboard. Array index numbers are from 0 to 49. If the user enters 50, there is no element in

students array with index 50. Here, JVM is unable to evaluate the result and thereby terminates the execution. Before terminating,

because Java is developed as a friendly language, it gives a message of the problem.

Every runtime problem, in Java, is represented by a class. For example, the problem of accessing an array element beyond the size of

array is represented by a class known as ArrayIndexOutOfBoundsException (from java.lang package). Similarly, division by zero problem is

represented by a class called ArithmeticException (from java.lang package).

Instead of writing this type theory more and more, to get a clear concept,

l give two programs one without handling exception and another with handling.

4)“Activity not found? Have you declared this activity in your manifest file”?

answer:-If you see that error occur after upgrading versions of IntelliJ IDEA or Android Studio, or after Generating a new APK,

you may need to refresh the IDE's cache.

File -> Invalidate Caches / Restart...

5)No such method getColor() found?

answer:-

The old method is deprecated starting in API 23, and the new method only exists in API 23+.

You are attempting to call the new method on a device running API <23.

You can either perform an API level check and call the appropriate method, or you can use ContextCompat.getColor(Context, int)

In your AppManifest.xml, make sure you have your code like this:

<uses-sdk

android:minSdkVersion="14"

android:targetSdkVersion="23"

android:maxSdkVersion="23" />

or for build.gradle:

android {

compileSdkVersion 23

buildToolsVersion "23.0.0"

defaultConfig {

applicationId "com.package.name"

minSdkVersion 14

targetSdkVersion 23

versionCode 1

versionName "1.0"

}

}from the support-v4 library.