1. **What are the four access modifiers available in Java and what is their significance in terms of class, method, and variable accessibility?**

* Access Modifiers are keywords that are used in object-oriented programming in order to specify the accessibility of the methods, classes, constructors, and other members of the class.

Four main types of access modifiers:

* Public 🡪 The code is accessible for all classes.
* Private 🡪 The code is only accessible within the declared class

* Protected 🡪 The code is accessible in the same package

and **subclasses**.

* Default 🡪 The code is only accessible in the same package.

This is used when you don’t specify a modifier.

1. Difference between Exception and Error?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Error** | **Exception** | | --- | --- | | Classified as an unchecked type | Classified as checked and unchecked | | It belongs to java.lang.error | It belongs to java.lang.Exception | | It is irrecoverable | It is recoverable | | It can't be occur at compile time | It can occur at run time compile time both | | Errors will not be known to the compiler. | Only checked exceptions are known to the compiler. | | Ex: OutOfMemoryError ,IOError | Ex: NullPointerException , SqlException | |  |
| 1. Difference between checked and unchecked exception  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | | **Checked Exception** | **Unchecked Exception** | | --- | --- | | Checked exceptions occur at compile time. | Unchecked exceptions occur at runtime. | | The compiler checks a checked exception. | The compiler does not check these types of exceptions. | | These types of exceptions can be handled at the time of compilation. | These types of exceptions cannot be a catch or handle at the time of compilation, because they get generated by the mistakes in the program. | | They are the sub-class of the exception class. | They are runtime exceptions and hence are not a part of the Exception class. | | Here, the JVM needs the exception to catch and handle. | Here, the JVM does not require the exception to catch and handle. | | Examples:  File Not Found Exception,  No Such Field Exception, Interrupted Exception,  No Such Method Exception,  Class Not Found Exception | Examples:  No Such Element Exception,  Undeclared Throwable, ExceptionEmpty Stack Exception, Arithmetic Exception,  Null Pointer Exception,  Array Index Out of Bounds Exception,  Security Exception | |  | |  |  | |  |  | |  | |  |  | |  |