**Task 11**

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

There are four access modifiers available in Java:

\*Public

\*Protected

\*Default

\*Private

**Class Access:**

**\*public:** Accessible from anywhere.

**\*protected,private,default:** Accessible only within the same package.

**Method Access:**

**\*public:** Accessible from anywhere.

**\*protected:** Accessible within the same package and by subclasses.

**\*private,default:** Accessible only within the same class.

**Variable Access:**

**\*public,protected,default:** Accessible within the same package.

**\*private:** Accessible only within the same class.

**2. What is the difference between Exception and error?**

**Exception:**

\*It can handle at runtime. (eg.IndexOutofBoundException)

\*Exceptions represent exceptional conditions that a program should anticipate and recover from. They are typically caused by conditions that occur at runtime, such as invalid user input, network failures, file I/O errors, etc.

\*Exceptions are further divided into two categories:

checked exceptions : Exception that occur at compile time

unchecked exceptions : Exception that occur at run time

\*Whenever the exception occurs at runtime is also known as runtime exception.

**Error:**

\*It cannot be handled at runtime. (eg. jdk crash,out of memory)

\*Errors represent serious, unrecoverable problems that are not usually handled by the program. They typically indicate problems that are outside the control of the application, such as system-level issues, hardware failures, out-of-memory errors, etc.

\*Errors are not meant to be caught or handled by normal application code.

**3. What is the difference between checked Exception and unchecked Exception?**

**Checked Exception:**

\*Checked exceptions are subclasses of exception but not subclasses of errors or runtime exceptions.

\*Checked exceptions must be either caught using a try-catch block or declared to be thrown in the method signature using the throws keyword.

\*If a method throws a checked exception, calling code must handle it or declare it to be thrown.

**Unchecked Exception:**

\*Unchecked exceptions are subclasses of RuntimeException and Error, or their subclasses.

\*Unchecked exceptions represent exceptional conditions that generally result from programming errors or conditions that the programmer could have avoided by better coding practices.

\*Unlike checked exceptions, unchecked exceptions are not required to be caught or declared. The compiler does not enforce handling of unchecked exceptions. This means that calling code is not obligated to catch or declare unchecked exceptions, although it can do so if needed.