**Task11 – Exception & Error**

**Q1. What are the four access modifiers available in java and what their significance in terms of class, method and variable accessibility.**

**Types of Access Modifiers:**

1. Default – No keyword required
2. Private
3. Protected
4. Public
5. **Default Modifiers:**
6. When no access modifier is specified for a class, method, or data member – It is said to be having the default access modifier by default.
7. The data members, classes, or methods that are not declared using any access modifiers that is having default access modifiers are accessible only within the same package.

**Example:**

package p1;

class Geek { // Default class

     void display() {  // Default Method

         System.out.println("Default Modifier!");

    }

}

1. **Private Modifiers:**
2. The private access modifier is specified using the keyword **private**.
3. The methods or data members declared as private are accessible only **within the class** in which they are declared.
4. Any other class of the same package will not be able to access these members.
5. Top-level classes or interfaces cannot be declared as private because

**Example:**

package p1;

class A  {

private void display() { // Private

        System.out.println("Hi all");

    }

}

class B

{

public static void main(String args[])

    {

        A obj = new A();

        // Trying to access private method

        // of another class

        obj.display();

    }

}

**//Result:**

error: display() has private access in A

obj.display();

**3. Protected Modifiers:**

1. The protected access modifier is specified using the keyword **protected**.
2. The methods or data members declared as protected are **accessible within the same package or subclasses in different packages.**

**Example:**

In this example, we will create two packages p1 and p2. Class A in p1 is made public, to access it in p2. The method display in class A is protected and class B is inherited from class A and this protected method is then accessed by creating an object of class B.

package p1;

// Class A

public class A

{

protected void display()

    {

        System.out.println("Hi all");

    }

}

package p2;

import p1.\*;

// Class B is subclass of A

class B extends A

{

public static void main(String args[])

{

    B obj = new B();

    obj.display();

}

}

**4. Public Modifiers:**

1. The public access modifier is specified using the keyword **public**.
2. The public access modifier has the widest scope among all other access modifiers.
3. Classes, methods, or data members that are declared as public are accessible from everywhere in the program. There is no restriction on the scope of public data members.

**Example:**

package p1;

public class A  {

public void display()  {

        System.out.println("Hi all");

    }

}

package p2;

import p1.\*;

class B {

    public static void main(String args[]){

        A obj = new A();

        obj.display();

    }

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Access Modifiers** | **Default** | **Private** | **Protected** | **Public** |
| **Same class** | Yes | Yes | Yes | Yes |
| **Same Package Subclass** | Yes | No | Yes | Yes |
| **Same Package Non-Subclass** | Yes | No | Yes | Yes |
| **Different Package Subclass** | No | No | Yes | Yes |
| **Different Package Non-Subclass** | No | No | No | Yes |

**Q2. What is the Difference between Exception & Error.**

In Java, errors and exceptions are both types of throwable objects, but they represent different types of problems that can occur during the execution of a program.

* An exception is an event that occurs during program execution, interrupting the normal flow of instructions.
* Exceptions can happen at compile time or run time & Error can happen at compile time.
* Exceptions are typically caused by issues in the code written by developers.
* Exceptions can be recovered using mechanisms like try-catch blocks or by declaring them with the throws keyword.
* Exceptionincludes**InputMismatchException**, **NullPointerException**, and **ArrayIndexOutOfBoundsException**.
* There are two types of exceptions:
  + **Checked exceptions:** These must be handled explicitly by the programmer (SQL, IO Exceptions).
  + **Unchecked exceptions:** These do not require explicit handling.
* Errors are usually caused by serious problems that are outside the control of the program, such as running out of memory or a system crash.
* In java, both Errors and Exceptions are the subclasses of java.lang.Throwable class.
* Errors are irrecoverable and cannot be caught or handled.
* Errors are always unchecked & Exceptions Include both Checked and unchecked.
* Errors include **OutOfMemoryError**, **StackOverflowError**, and **LinkageError**.
* Errors are mostly caused by the environment in which program is running while in Exceptions Program itself is responsible for causing exceptions.

**Q3. What is the Difference Between checked Exception and unchecked Exception.**

Exception is an unwanted or unexpected event, which occurs during the execution of a program, i.e. at run time, that disturbs the normal flow of the program’s instructions.

**Two types of Exception:**

1. Checked exceptions
2. Unchecked exceptions

* Checked exceptions occur at compile time & Unchecked exceptions occur at runtime.
* The compiler checks a checked exception & compiler does not check these types of exceptions in unchecked exceptions.
* Checked types of exceptions can be handled at the time of compilation while unchecked exceptions cannot be a catch or handle at the time of compilation, because they get generated by the mistakes in the program.
* Checked exceptions are the sub-class of the exception class & unchecked exceptions are runtime exceptions so it is not a part of the Exception class.
* In Checked exceptions JVM needs the exception to catch and handle.
* In unChecked exceptions JVM does not require the exception to catch and handle.

**Examples of Checked exceptions:**

* File Not Found Exception
* No Such Field Exception
* Interrupted Exception
* No Such Method Exception
* Class Not Found Exception

**Examples of Unchecked Exceptions:**

* No Such Element Exception
* Undeclared Throwable Exception
* Empty Stack Exception
* Arithmetic Exception
* Null Pointer Exception
* Array Index Out of Bounds Exception
* Security Exception