13*.Program for throw and throws in Java

The throw keyword is used to **explicitly** throw an exception, while throws is used to **declare** exceptions that a method might throw.

---

1. Using throw (Manually throwing an exception)

```
public class ThrowExample {
    static void checkAge(int age) {
        if (age < 18) {
            throw new ArithmeticException("Access denied - You must be 18 or older.");
        } else {
            System.out.println("Access granted - You are old enough!");
        }
    }

    public static void main(String[] args) {
        checkAge(16); // Throws an exception
    }
}
```

Explanation:

- throw manually throws an ArithmeticException if age is less than 18.
- The exception **stops** execution if triggered.

Expected Output:

```
Exception in thread "main" java.lang.ArithmeticException: Access denied - You must be 18 or older.
```

---

2. Using throws (Declaring an exception)

```
import java.io.*;


public class ThrowsExample {
   static void readFile() throws IOException {
       throw new IOException("File not found!"); // Simulating file read failure
   }


   public static void main(String[] args) {
       try {
           readFile(); // This method declares that it may throw IOException
       } catch (IOException e) {
           System.out.println("Caught Exception: " + e.getMessage());
       }
   }
}
```

Explanation:

- throws IOException declares that readFile() may throw an exception.
- The exception is **caught** in main() inside a try-catch.

Expected Output:

```
Caught Exception: File not found!
```

Key Differences:

Keyword	Purpose	Example
throw	Manually throws an exception	throw new ArithmeticException("Error!");
throws	Declares that a method may throw exceptions	void myMethod() throws IOException