Throw Keyword

Definition:

The Java *throw* keyword is used to explicitly throw an exception. We can throw either checked or unchecked exceptions in java by *throw* keyword. The *throw* keyword is mainly used to throw custom exceptions.

Syntax:

```
throw exception
```

Need for *throw* keyword:

The *throw* keyword throws custom error and stops execution of remaining statements in the block.

Working of *throw* keyword:

The flow of execution of the program stops immediately after the throw statement is executed and the nearest enclosing **try** block is checked to see if it has a **catch** statement that matches the type of exception. If it finds a match, control is transferred to that statement otherwise the next enclosing **try** block is checked and so on. If no matching **catch** is found then the default exception handler will halt the program.

Examples:

1. Throw keyword without handling exception:

```
public class Main
{
    static void validate(int age)
    {
        if(age<18)
            throw new ArithmeticException("not valid");
        else
            System.out.println("welcome to vote");
    }
    public static void main(String args[])
    {
       validate(13);
       System.out.println("rest of the code...");
    }
}
```

OUTPUT

```
Exception in thread main java.lang.ArithmeticException:not valid
```

Explanation:

Exception is thrown explicitly using the *throw* keyword. There is no try-catch block to handle exceptions. So error occurs.

2. Throw keyword with handling exception:

```
public class Main
{
    static void validate(int age)
    {
        try
        {
            if(age<18)
                throw new ArithmeticException("not valid");
            else
                System.out.println("welcome to vote");
        }
        catch(ArithmeticException e)
        {
            System.out.println(e);
        }
    }
    public static void main(String args[])
    {
       validate(13);
       System.out.println("rest of the code...");
    }
}
```

OUTPUT

```
java.lang.ArithmeticException:not valid
rest of the code...
```

Explanation:

Exception is thrown explicitly using the *throw* keyword. There is a catch block to handle exceptions. So the program runs smoothly without interruption. Remaining statements are executed.

Important points to remember:

- ➔ The *throw* keyword must be followed by an instance of Throwable class or one of its subclasses.
- ➔ When using the *throw* keyword to throw a checked exception from within a method, the method must either:
    - ◆ Declares the throws clause followed by the exceptions thrown by the *throw* statements, or:
    - ◆ Catch the exceptions thrown by the throw statements.

Difference between throw and throws keyword:

S. No	Keys	***throws***	***throw***
1	Definition	*Throws* is a keyword used in the method signature used to declare an exception which might get thrown by the function while executing the code.	*Throw* is a keyword which is used to throw an exception explicitly in the program inside a function or inside a block of code.
2	Internal implementation	we can declare multiple exceptions with *throws*	Internally, *throw* is implemented as it is

		keyword that could get thrown by the function where *throws* keyword is used.	allowed to throw only a single exception at a time i.e we cannot throw multiple exceptions with a *throw* keyword.
3	Type of exception	Using *throws* keyword both checked and unchecked exceptions can be declared and for the propagation checked exception must use throws keyword followed by specific exception class name.	With *throw* keyword we can propagate only unchecked exceptions i.e checked exceptions cannot be propagated using throw.
4	Syntax	*throws* keyword is followed by exception class names.	*throw* keyword is followed by the instance variable.
5	Declaration	*throws* keyword is used with the method signature.	In order to use the *throw* keyword we should know that throw keyword is used within the method.