Throw Keyword

Throw in Java is a keyword that is used to throw a built-in exception or a custom exception explicitly or manually. Using throw keyword, we can throw either checked or unchecked exceptions in Java programming.

When an exception occurs in the try block, throw keyword transfers the control of execution to the caller by throwing an object of exception.

Key points of Throw keyword:

1. In Java exception handling, we use throw keyword to throw a single exception explicitly. It is followed by an instance variable.

2. Using throw keyword, we can throw either checked or unchecked exception in Java.

3. The keyword throw raises an exception by creating a subclass object of Exception explicitly.

4. We mainly use throw keyword to throw custom exception on the basis of some specified condition.

5. We use keyword throw inside the body of method or constructor to invoke an exception.

6. With the help of throw keyword, we cannot throw more than one exception at a time.

```
public class ThrowTest1 {
public static void main(String[] args) {

// Declaring a try-catch block.
 try
 {
// Creating an object of ArithmeticException class.
   ArithmeticException a = new ArithmeticException("Hello from throw"); // Line 7
   throw a; // Exception thrown explicitly. // Line 8

 // Line 7 and 8 can be written also in one line like this:
 // throw new ArithmeticException("Hello from throw");
}
catch(ArithmeticException ae)
{
   System.out.println("ArithmeticException caught: \n" +ae);
   System.out.println(ae.getMessage());
```

```
  }
 }
}
```

```
public class ThrowTest2 {
public static void main(String[] args)
{
 int x = 20;
 int y = 0;
 try
 {
   int z = x/y; // Exception occurred. // Line 9
   System.out.println("Result: " +z); // Line 10
   throw new ArithmeticException();
 }
 catch(ArithmeticException ae) {
     System.out.println("Exception caught: \n" +ae);
 }
}
}
```

```
public class ThrowTest3 {
public static void main(String[] args)
{
 int x = 20;
 int y = 0;

 try
 {
   int z = x/y;
   throw new ArithmeticException();
   System.out.println("Result: " +z); // Unreachable code.
 }
 catch(ArithmeticException ae) {
   System.out.println("Exception caught: \n" +ae);
  }
 }
}
```

```
public class ThrowTest4 {
public static void main(String[] args)
{
  int num = 1;
  for(num = 1; num <= 10; num++) {
  try
  {
    if(num == 5)
        throw new ArithmeticException("ArithmeticException");
    else if(num < 2)
        throw new RuntimeException("RuntimeException");
```

```
     else if(num > 9)
         throw new NullPointerException("NullPointerException");
   }
   catch(Exception e)
   {
      System.out.println("Caught an exception");
      System.out.println(e.getMessage());
   }
  }
  }
}
```