

# Try Catch Block

## Five keywords to handle Exception in Java

---

- try
- catch
- finally
- throw
- throws

## Try Block in Java

---

A keyword "try" is a block of code or statements that might throw an exception. That's why a try block is also known as exception generated block.

The Java code that may generate an exception during the execution of program, must be placed within a try block.

That is, we should place exception generated code (risky code) inside try block. We should not keep normal code inside try block.

The three possible forms of try block are as follows:

- try-catch: A try block is always followed by one or more catch blocks.
- try-finally: A try block followed by a finally block.
- try-catch-finally: A try block followed by one or more catch blocks followed by a finally block.

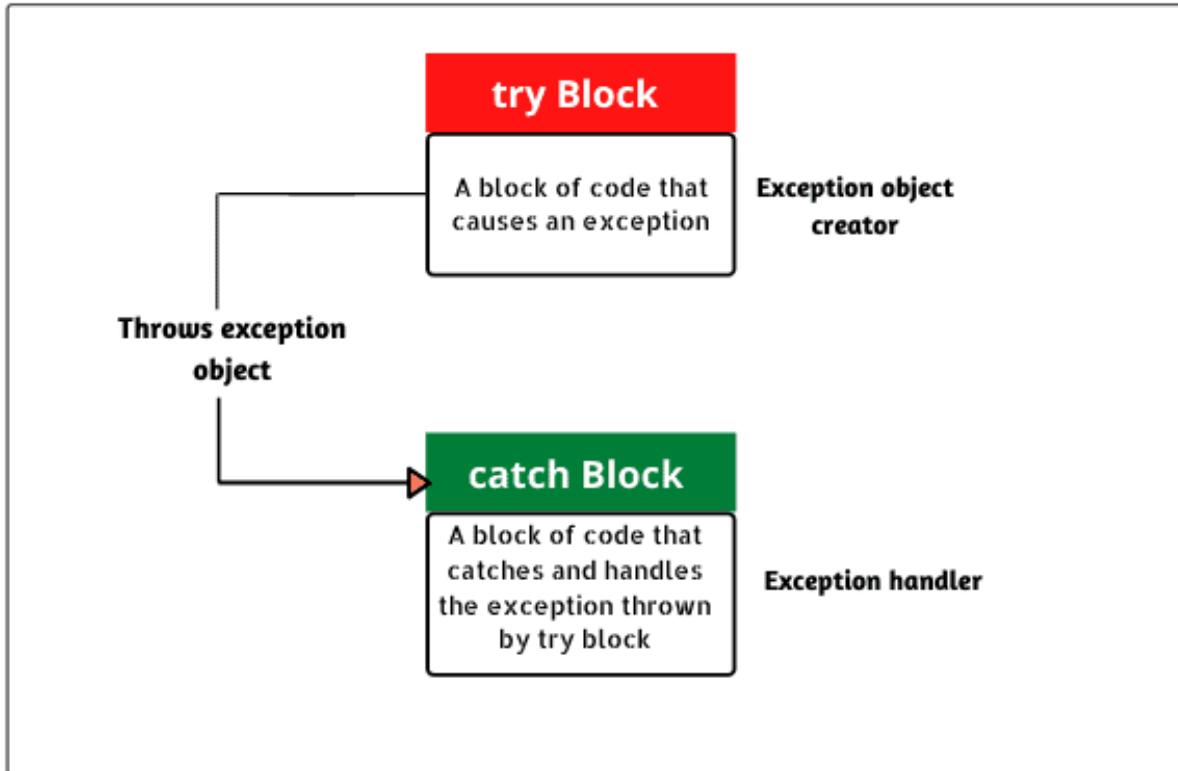
## Catch Block in Java

---

A keyword "catch" is a block of code that handles the exception thrown by the try block. That's why it is also known as exception handler block.

A catch block that catches an exception, must be followed by try block that generates an exception.

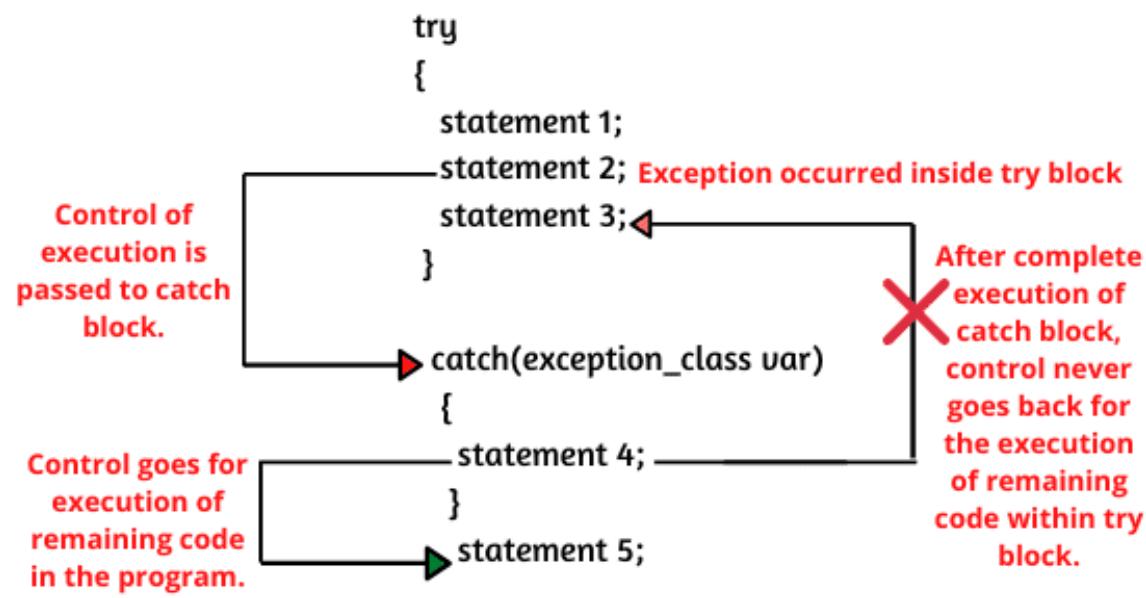
## Exception Handling Mechanism using Try-Catch block



## Rules for using Try-Catch block in Java

There are some rules for using try-catch block in java program. They are as follows:

1. Java try-catch block must be within a method.
2. A try block can not be used without a catch or finally block. It must be followed by at least one catch block otherwise, the compilation error will occur.
3. A catch block must be followed by try block. There should not be any statement between the end of try block and the beginning of catch block.
4. A finally block cannot come before catch block.



```

public class TryCatchEx {
public static void main(String[] args)
{
    System.out.println("11");
    System.out.println("Before divide");
    int x = 1/0; // Exception occurred.
    System.out.println("After divide");
    System.out.println("22");
}
}

```

```

public class TryCatchEx1 {
public static void main(String[] args)
{
    System.out.println("11");
    System.out.println("Before divide");
}
}

```

```
// Applying try-catch block to handle exception.  
try  
{  
    int x = 1/0;  
    System.out.println("After divide");  
}  
catch(ArithmaticException ae) // Here, ae is a reference variable of exception object.  
{  
    System.out.println("A number cannot be divided by zero");  
}  
System.out.println("22");  
}  
}
```

```
public class TryCatchEx2 {  
public static void main(String[] args)  
{  
    System.out.println("111");  
    try  
{  
        int x = 12/0; // exception occurred.  
        System.out.println("Result of x: " +x);  
        System.out.println("333");  
    }  
    catch(ArithmaticException ae)  
{  
        System.out.println("Hello world");  
    }  
    System.out.println("444");  
}
```

```
public class TryCatchEx3 {  
    public static void main(String[] args)  
    {  
        int x = 100, y = 0;  
        try  
        {  
            System.out.println("111");  
            int z = x/y;  
            System.out.println("Result of z: " +z);  
        }  
        catch(ArithmeticException ae)  
        {  
            System.out.println("Hello Java");  
        }  
        System.out.println("333");  
    }  
}
```

```
public class TryCatchEx4  
{  
    int x = 30, y = 0;  
    void divide()  
    {System.out.println("I am in method");  
        try  
        {  
            System.out.println("I am in try block");  
            int z = x/y;  
            System.out.println("Result of z: " +z);  
        }  
        catch(NullPointerException np)  
        {System.out.println("I am in catch block");}  
    }  
    public static void main(String[] args)  
    {
```

```
TryCatchEx4 obj = new TryCatchEx4();
System.out.println("I am in main method");
obj.divide();
}
```

```
public class TryCatchEx5 {
public static void main(String[] args)
{
try
{
    System.out.println("111");
    System.out.println("222");
}
catch(ArithmeticException ae)
{
    System.out.println("333");
}
System.out.println("444");
}
}
```

```
public class TryCatchEx6 {
public static void main(String[] args)
{
    System.out.println("111");
    try
    {
        System.out.println("222");
        int y = 1/0;
    }
    catch(ArithmeticException e)
    {
        try
```

```
{  
    System.out.println("Hello");  
    int x = 20/0;  
}  
catch(NullPointerException np)  
{  
    System.out.println("333");  
}  
}  
}  
System.out.println("444");  
}  
}
```

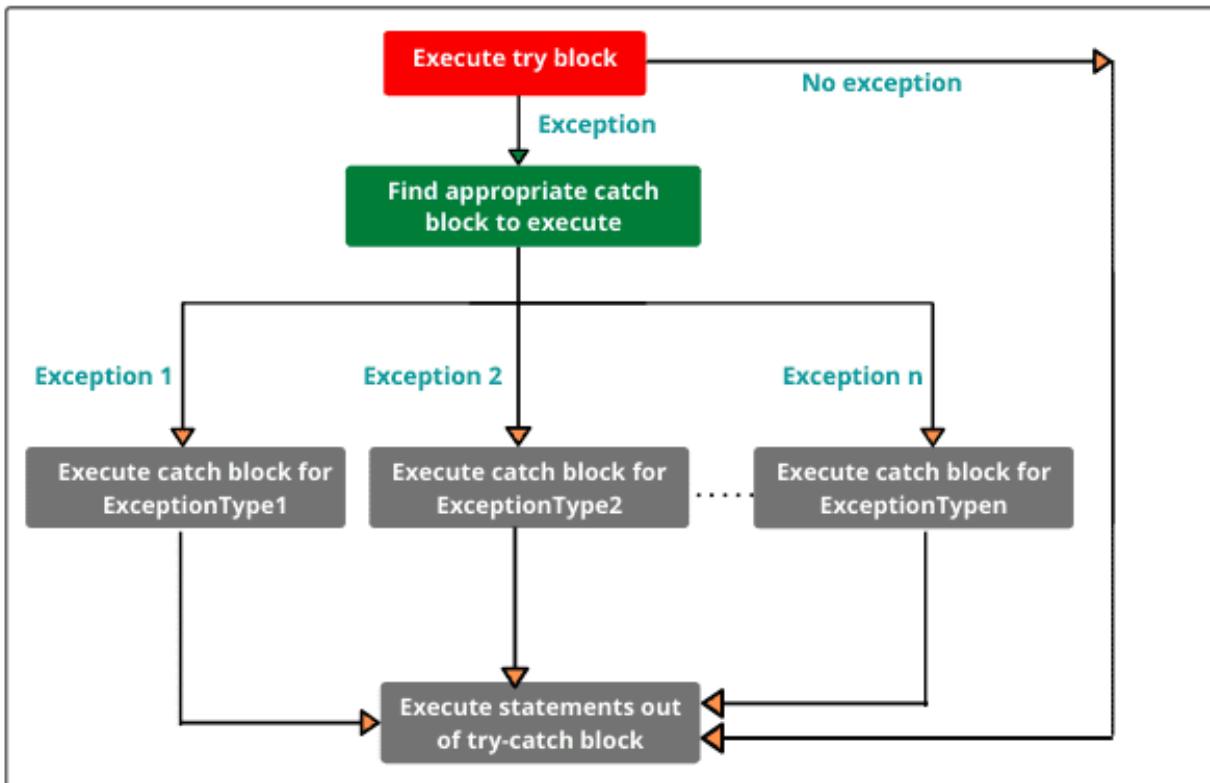
```
public class TryCatchEx7 {  
    public static void main(String[] args)  
    {  
        try  
        {  
            int a[] = {20, 30, 40, 50};  
            a[10] = 5;  
        }  
        catch(ArrayIndexOutOfBoundsException a)  
        {  
            System.out.println("Array Index Out Of Bounds Exception");  
        }  
    }  
}
```

```
public class TryCatchEx8 {  
    public static void main(String[] args)  
    {  
        try  
        {  
            // This method returns the Class object associated with the class or interface  
        }  
    }  
}
```

with the given string name.

```
Class c = Class.forName("ArithmeticException");
}
catch(ClassNotFoundException cn)
{
    System.out.println(cn.getMessage());
}
}
```

```
public class TryCatchEx9 {
public static void main(String[] args)
{
try
{
    String input = "Puneet";
    int a = Integer.parseInt(input);
    System.out.println("Value of a: " +a);
}
catch(NumberFormatException n)
{
    System.out.println(n.getMessage() + " is not an integer.");
}
}}
```



```

package exceptionHandling;
public class MultiCatchEx1
{
    public static void main(String[] args)
    {
        try
        {
            // Creating an array of six integer elements.
            int arr[] = new int[6];
            arr[3] = 20/0; // Exception occurred.
            System.out.println("I am in try block");
        }
        catch(ArithmeticException ae)
        {
            System.out.println("A number cannot be divided by zero, illegal operation in java");
        }
    }
}

```

```
catch(ArrayIndexOutOfBoundsException e)
{
    System.out.println("Accessing array element outside of specified limit");
}
catch(Exception e)
{
    System.out.println(e.getMessage());
}
System.out.println("I am out of try-catch block");
}
```

```
package exceptionHandling;
public class MultiCatchEx2
{
public static void main(String[] args)
{
    String s = "Puneet";
    int a[] = {0, 1, 2, 3, 4, 5};
    try
    {
        s = null;
        int sLength = s.length();
        System.out.println("String length: " +sLength);
        int b = 6;
        System.out.println(a[b]);
    }
    catch(NullPointerException npe)
    {
        System.out.println("Exception is caught");
        System.out.println(npe.toString());
    }
    catch(ArrayIndexOutOfBoundsException aie)
    {
        System.out.println("Exception is caught");
    }
}
```

```
        System.out.println(aie.toString());
    }
}
}
```

```
package exceptionHandling;
import java.util.Scanner;
public class MultiCatchEx3
{
    public static void main(String[] args)
    {
        int x, y;
        // Creating a scanner object.
        Scanner sc = new Scanner(System.in);

        // Applying try-catch block.
        try
        {
            System.out.println("Enter your first number");
            x = Integer.parseInt(sc.nextLine());
            System.out.println("Enter your second number");
            y = Integer.parseInt(sc.nextLine());

            int z = x / y;
            System.out.println("z = " +z);
        }
        catch(ArithmaticException ae)
        {
            System.out.println("A number cannot be divided by 0, Illegal operation in Java");
            System.out.println("Exception thrown: " +ae);
        }
        catch(NumberFormatException nfe)
        {
            System.out.println("Invalid data types are entered, number must be an inte
```

```

ger.");
    System.out.println("Exception thrown: " +nfe);
}
catch(RuntimeException re)
{
    System.out.println("Exception thrown: " +re);
}
System.out.println("Out of try-catch block");
}
}

```

## What is Unreachable Catch Block Error?

```

package exceptionHandling;
import java.util.Scanner;
public class MultiCatchEx4 {
public static void main(String[] args)
{
int x, y;
Scanner sc = new Scanner(System.in);

try
{
    System.out.println("Enter your first number");
    x = Integer.parseInt(sc.nextLine());
    System.out.println("Enter your second number");
    y = Integer.parseInt(sc.nextLine());

    int z = x / y;
    System.out.println("z = " +z);
}
catch(RuntimeException re)
{
    System.out.println("Exception thrown: " +re);
}

```

```
}

catch(ArithmetricException ae) // Unreachable catch block error.
{
    System.out.println("Exception thrown: " +ae);
}
catch(NumberFormatException nfe) // Unreachable catch block error.
{
    System.out.println("Exception thrown: " +nfe);
}
}
```