INDEX :

Serial No.	*Type Of Code*	*Page No.*
1	Exception Handling	2
2	Handling Specific Excep.	4
3	Nested Try & Catch Block	6
4	Exception Class	7
5	throw & throws	9
6	finally Block	11

Exception Handling :

```java
import java.util.Scanner;

public class ZI_ExcpHandlingTryAndCatch{

    public static void main(String[] args) {

        int num1, num2, div, flag;

        Scanner sc = new Scanner(System.in);


        System.out.println("By using try and catch method :-");

        do{

            flag = 0;

            System.out.println("Enter two numbers here :-");

            num1 = sc.nextInt();

            num2 = sc.nextInt();

            try{

                // We put those statements in try on which we thought that may gives error.

                div = num1/num2;

                System.out.println("The value of division is :  " + div);

            }

            catch(Exception e){

                System.out.println("Failed to divide");

                System.out.println("Reason :  " + e);

                System.out.println("Please enter valid numbers");

                flag = 1;

            }

        }while(flag == 1);


        // System.out.println("Without using try and catch method :-");

        // div = num1/num2;

        // System.out.println("The value of division is :  " + div); // This will throw error.


        System.out.print("If we use try and catch methods then it execute next lines ");

        System.out.println("of code also after giveing catch exception");

        sc.close();

    }

}
```

Output :-

```
PS D:\11. Tutorial of Java> cd "d:\11. Tutorial of Java\" ; if ($?) { javac ZI_ExcpHandlingTryAndCatch.
By using try and catch method :-
Enter two numbers here :-
56
0
Failed to divide
Reason :  java.lang.ArithmeticException: / by zero
Please enter valid numbers
Enter two numbers here :-
5
1
The value of division is :  5
If we use try and catch methods then it execute next lines of code also after giveing catch exception
PS D:\11. Tutorial of Java>
```

Handling Specific Exception :

```java
import java.util.Scanner;

public class ZJ_HandlingSpecificException{

    public static void main(String[] args) {

        System.out.print("Enter the number of terms you want in the array :  ");

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int [] arr = new int[n];

        System.out.println("Enter the values in the array here :-");

        for(int i=0; i<n; i++){

            arr[i] = sc.nextInt();

        }

        System.out.println("Values entered in the array is as follow :-");

        for(int element : arr){

            System.out.print(element + "\t");

        }


        System.out.println();

        System.out.println("Enter the array index on which you want to perform operation");

        int idx = sc.nextInt();

        System.out.println("Enter the number you want to divide the array value with");

        int num = sc.nextInt();


        try{

            System.out.println("The value at array on entered index is :  " + arr[idx]);

            System.out.println("The value of "+arr[idx]+" / "+num+" is = "+arr[idx]/num);

        }

        catch(ArithmeticException e){

            System.out.println("Arithmetic Exception occured..!");

            System.out.println(e);

        }

        catch(ArrayIndexOutOfBoundsException e){

            System.out.println("Exception :-\nArray index is out of boundation..!");

            System.out.println(e);

        }
```

```
        catch(Exception e){

            System.out.println("Some exception occured..!");

            System.out.println(e);

        }


        sc.close();

    }

}
```

Output :-

```
PS D:\11. Tutorial of Java> cd "d:\11. Tutorial of Java\" ; if ($?) { javac 
Enter the number of terms you want in the array :  4
Enter the values in the array here :-
2
3
5
6
Values entered in the array is as follow :-
2       3       5       6
Enter the array index on which you want to perform operation
5
Enter the number you want to divide the array value with
1
Exception :-
Array index is out of boundation..!
java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 4
PS D:\11. Tutorial of Java>
```

Nested Try & Catch Block :

```java
public class ZK_NestedTryAndCatch{

    public static void main(String[] args){

        int [] marks = new int[5];

        marks[0] = 91;

        marks[1] = 92;

        marks[2] = 94;

        marks[3] = 96;

        marks[4] = 98;

        try{

            System.out.println("Welcome to my program");

            System.out.println(marks[10]); // After detecting this program shows its excep.

            try{

                System.out.println(marks[9]);

            }

            catch(ArrayIndexOutOfBoundsException e){

                System.out.println("Sorry this index doesn't exist");

                System.out.println("Exception in level 2");

            }

        }

        catch(Exception e){

            System.out.println("Exception in level 1");

        }


    }

}
```

Output :-

```
PS D:\11. Tutorial of Java> cd
Welcome to my program
Exception in level 1
PS D:\11. Tutorial of Java>
```

Exception Class :

```java
import java.util.Scanner;

class MyException extends Exception{

    @Override

    public String toString(){

        return "I'm toString()";

    }

    @Override

    public String getMessage(){

        return "I'm getMessage()";

    }

}

public class ZL_ExceptionClass{

    public static void main(String[] args){

        System.out.print("Enter a number here :  ");

        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();

        sc.close();

        if(a<10){

            try{

            // 'throw' keyword is used to throw an exception explicetly by the programmer.

                throw new MyException();

            // throw new ArithmeticException("This is an Exception"); //use like this also

            }

            catch(Exception e){

                System.out.println(e.getMessage());

                System.out.println(e.toString());

                System.out.println(e); // By default it runs toString method.

                System.out.println("Finshed..!");

                e.printStackTrace(); //-> shows where error is occured and in which line.

            }

            System.out.println("if condition finished here..!");

        }

    }

}
```

Output :-

```
PS D:\11. Tutorial of Java> cd "d:\11. Tutorial of Java\" ; if ($?)
Enter a number here :  51
PS D:\11. Tutorial of Java> cd "d:\11. Tutorial of Java\" ; if ($?)
Enter a number here :  5
I'm getMessage()
I'm toString()
I'm toString()
Finshed..!
I'm toString()
        at ZL_ExceptionClass.main(ZL_ExceptionClass.java:24)
if condition finished here..!
PS D:\11. Tutorial of Java>
```

throw & throws :

```java
class NegativeRadiusException extends Exception{

    @Override

    public String toString() {

        return "Radius cann't be negative..!";

    }

}


public class ZM_throwVSthrows{

    static void CArea(double r) throws NegativeRadiusException{

        if(r<0){

            try{

                throw new NegativeRadiusException();

            }

            catch(Exception e){

                System.out.println(e.toString());

            }

        }

        else{

            System.out.print("Area of given circle with radius " + r + " is :  ");

            System.out.println(Math.PI * r * r);

        }

    }


    static int div(int a, int b) throws ArithmeticException{

        // Made by me

        int result = a/b;

        return result;

    }

    public static void main(String[] args) {

        // You - uses div function created by me

        try{

            int c = div(6, 0);

            System.out.println("Result is :  " + c);

        }
```

```
        catch(Exception e){
            System.out.println("This is Arithmetic Exception");
            System.out.println(e);
        }

        // double ar = CArea(5); // --> throw error as it may throw exception and we must
        // have to use try and catch block, as we use 'throws' keyword.
        try{
            CArea(5);
            CArea(-1);
        }
        catch(Exception e){
            System.out.println("Exiting..!");
        }

    }
}
```

Output :-

```
PS D:\11. Tutorial of Java> cd "d:\11. Tutorial of Java\" ;
This is Arithmetic Exception
java.lang.ArithmeticException: / by zero
Area of given circle with radius 5.0 is :  78.53981633974483
Radius cann't be negative..!
PS D:\11. Tutorial of Java>
```

finally Block :

```java
public class ZN_FinallyBlock{

    static int greet(){

        try{

            int a = 50;

            int n = 2;

            int v = a/n;

            return v;

        }

        catch(Exception e){

            System.out.println(e);

        }

        finally{ // statement in 'finally' runs even after function return value.

            System.out.println("Cleaning up resource...");

            System.out.println("This is end of this function");

        }

        System.out.print("Hey this will not going to run in main program if");

        System.out.println("Exception doesn't occur");

        // As it is out of 'finally' block and written after returning some value..!


        return 0;

    }

    public static void main(String[] args){

        int r = greet();

        System.out.println("Collected value is :  " + r);


        int a = 6;

        int b = 9;

        while(true){

            try{

                System.out.println("The value of " + a + "/" + b + " is :  " + (a/b));

            }

            catch(Exception e){

                System.out.println(e);

                break;
```

```
            }
            finally{
                System.out.println("I'm finally for value of b = " + b);
            }
            System.out.println("This is end of iteration");
            /*
            for b = 0 , when break of this loop is occur then still finally will going to
            run while other statement after break of loop doesn't.
            */
            b--;
        }

        try{
            System.out.println(5/0);
        }
        finally{
            System.out.print("After 'try' method it is not neccessory to use catch ");
            System.out.println("while we must have to use 'finally'");
            System.out.println("But then this program will going to throw error..!");
            System.out.println("As we don't handle the exception");
        }

    }
}
```

Output :-

```
PS D:\11. Tutorial of Java> cd "d:\11. Tutorial of Java\" ; if ($?) { javac ZN_FinallyBlock
Cleaning up resource...
This is end of this function
Collected value is :  25
The value of 6/9 is :  0
I'm finally for value of b = 9
This is end of iteration
The value of 6/8 is :  0
I'm finally for value of b = 3
This is end of iteration
The value of 6/2 is :  3
I'm finally for value of b = 2
This is end of iteration
The value of 6/1 is :  6
I'm finally for value of b = 1
This is end of iteration
java.lang.ArithmeticException: / by zero
I'm finally for value of b = 0
After 'try' method it is not neccessory to use catch while we must have to use 'finally'
But then this program will going to throw error..!
As we don't handle the exception
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at ZN_FinallyBlock.main(ZN_FinallyBlock.java:48)
PS D:\11. Tutorial of Java>
```