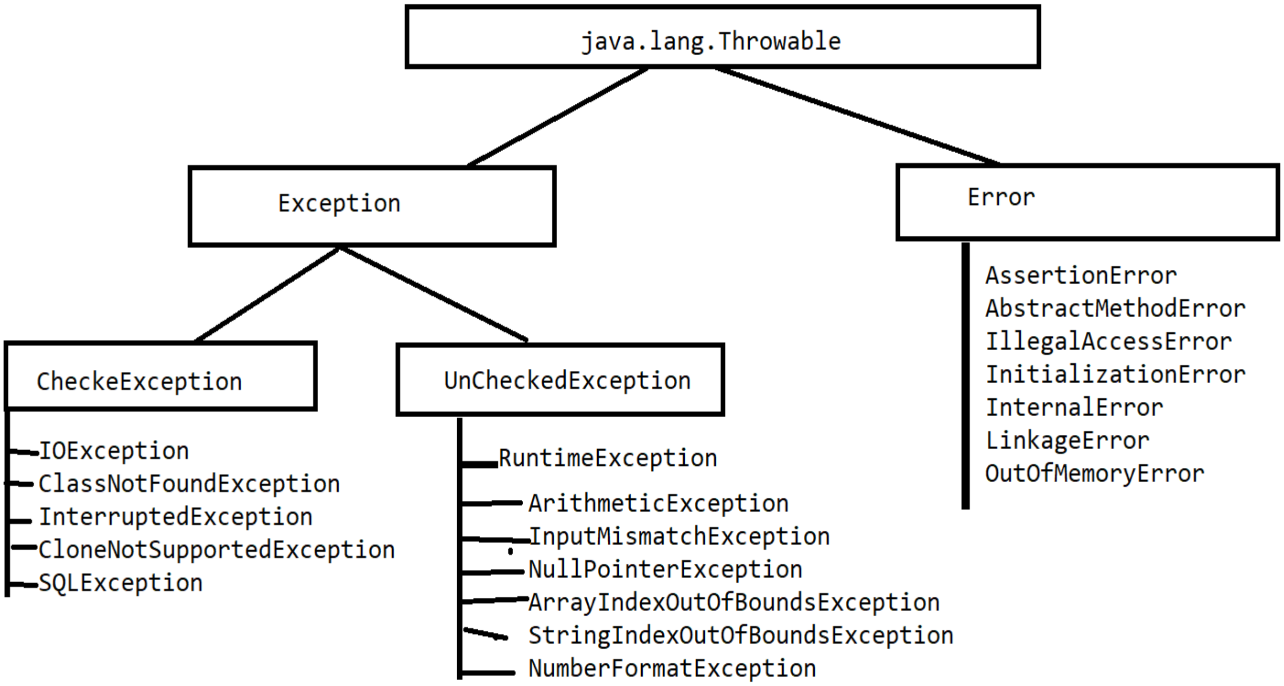
## Hierarchy of Exception Handling

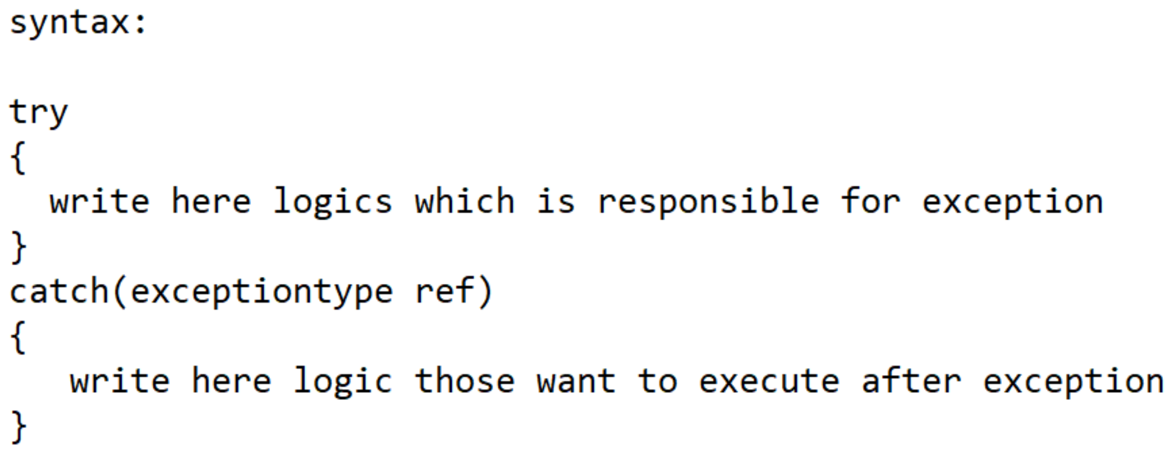


## Keywords for Exception Handling

If we want to work with exception in java we have the major five keywords provided by java to us.

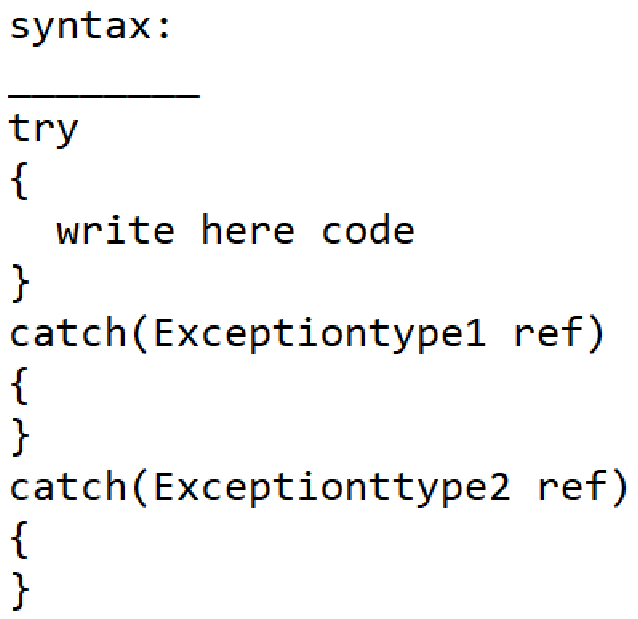
**try :** try is block in java which is used for write the code in which exception may be occur or normally we write code in try block which is responsible for generate the exception in program. When exception generate in try block then JVM create the one error object and hand over to catch for further execution.

**catch:** catch is block which always execute when exception generate in try block Normally Catch block is used for execute the logic those want to execute after exception or write the logic those want to execute after exception as well as we can show the exception message using catch block. Catch block hold the error object thrown by try block and display it.



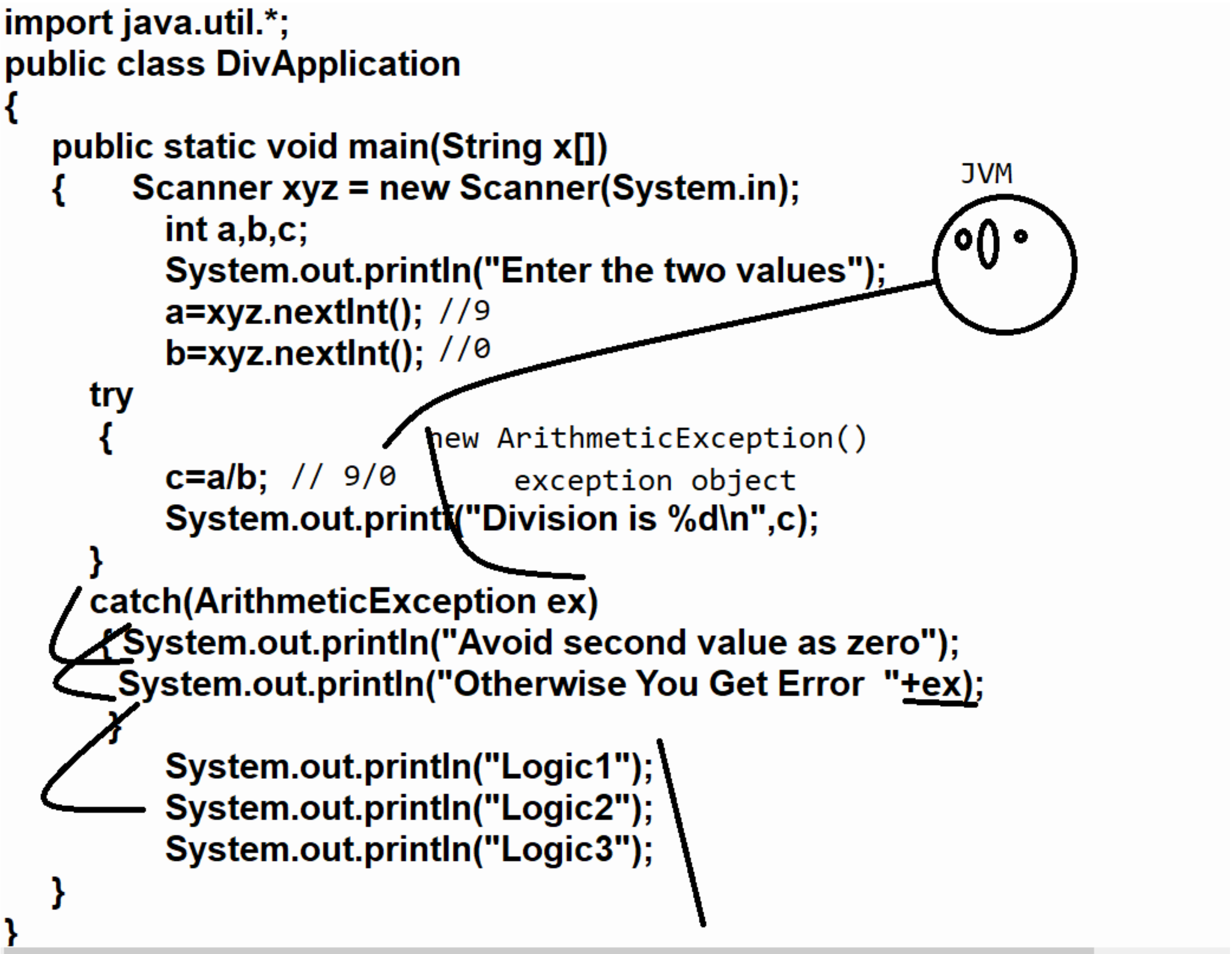
```
syntax:

try
{
  write here logics which is responsible for exception
}
catch(exceptiontype ref)
{
   write here logic those want to execute after exception
}
```

**Note: single try can have more than one catch block.**



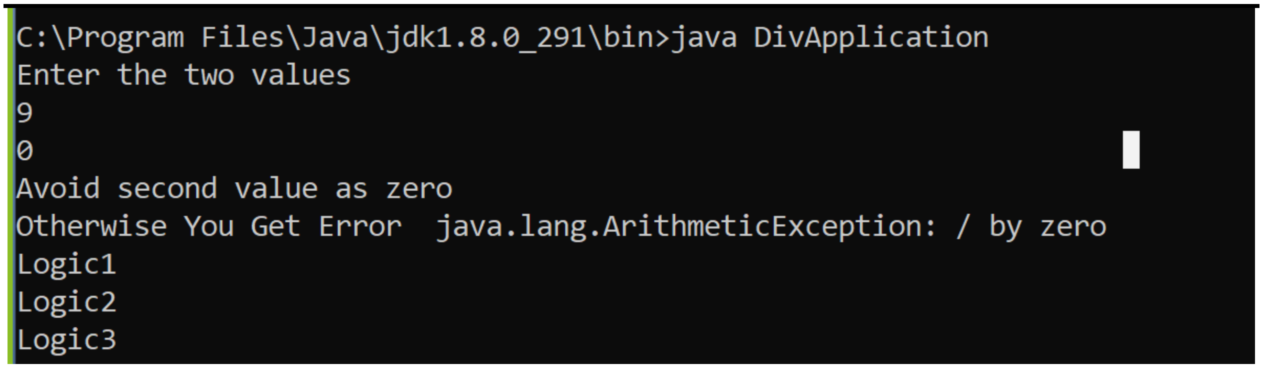
```
syntax:
_________
try
{
  write here code
}
catch(Exceptiontype1 ref)
{
}
catch(Exceptionttype2 ref)
{
}
```

**Example**



```
import java.util.*;
public class DivApplication
{
    public static void main(String x[])
    {      Scanner xyz = new Scanner(System.in);
             int a,b,c;
             System.out.println("Enter the two values");
             a=xyz.nextInt(); //9
             b=xyz.nextInt(); //0
       try
        {                          new ArithmeticException()
             c=a/b;  // 9/0            exception object
             System.out.printf("Division is %d\n",c);
       }
       catch(ArithmeticException ex)
         {System.out.println("Avoid second value as zero");
         System.out.println("Otherwise You Get Error  "+ex);
         }
             System.out.println("Logic1");
             System.out.println("Logic2");
             System.out.println("Logic3");
    }
}
```

**Output**



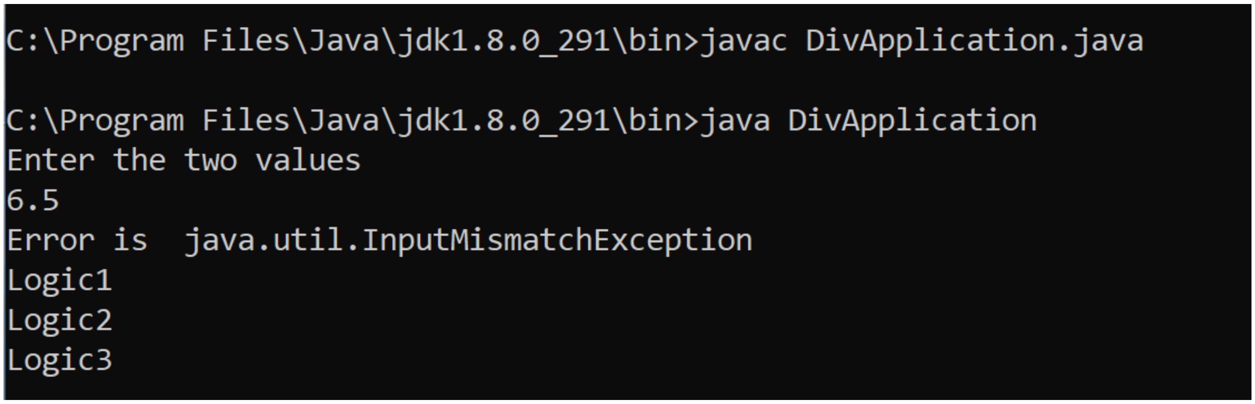
```
C:\Program Files\Java\jdk1.8.0_291\bin>java DivApplication
Enter the two values
9
0
Avoid second value as zero
Otherwise You Get Error  java.lang.ArithmeticException: / by zero
Logic1
Logic2
Logic3
```

## Single Try can have multiple catch blocks.

**import java.util.*;**

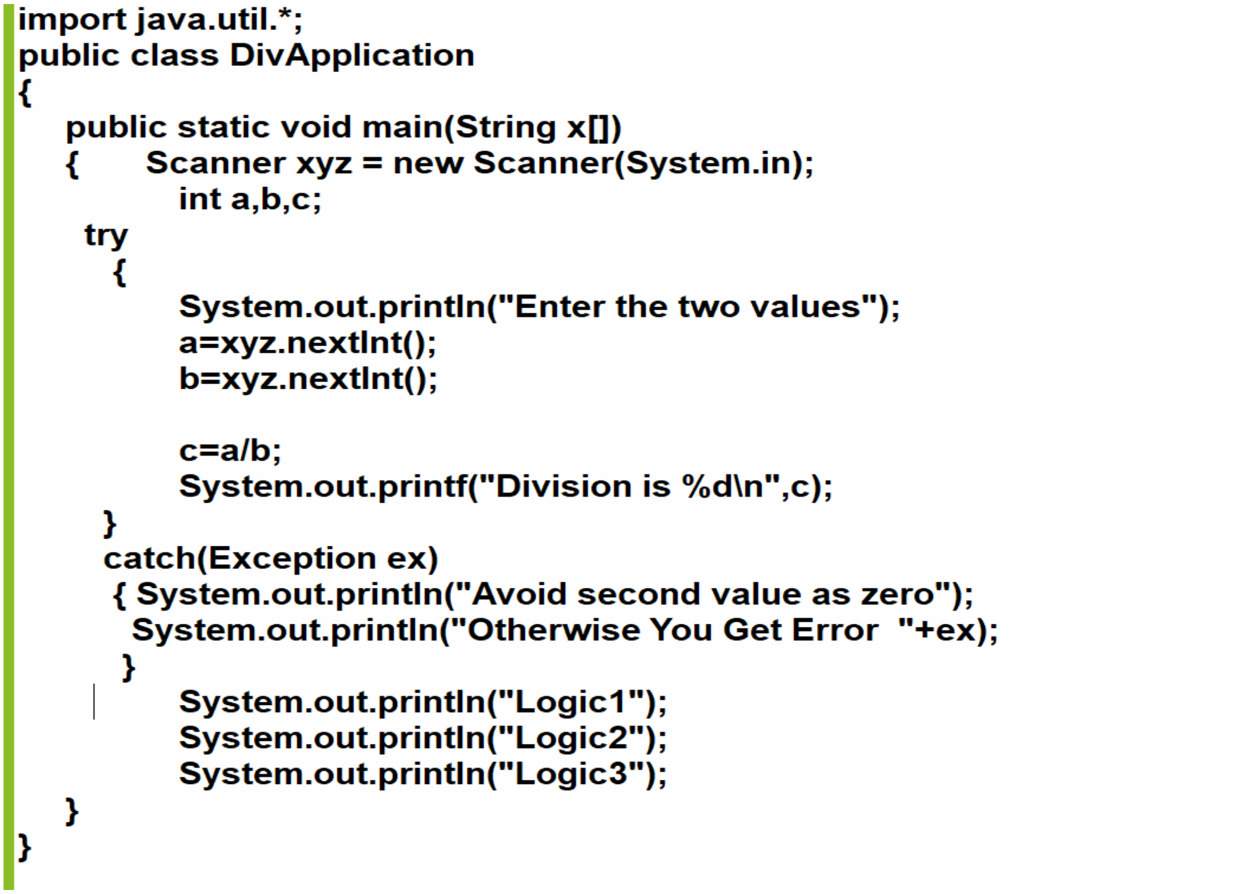
```
public class DivApplication
{
  public static void main(String x[])
  {     Scanner xyz = new Scanner(System.in);
          int a,b,c;
    try
      {
          System.out.println("Enter the two values");
          a=xyz.nextInt();
          b=xyz.nextInt();
          c=a/b;
          System.out.printf("Division is %d\n",c);
      }
      catch(ArithmeticException ex)
      { System.out.println("Avoid second value as zero");
        System.out.println("Otherwise You Get Error  "+ex);
       }
      catch(InputMismatchException ex)
     {
         System.out.println("Error is  "+ex);
      }
          System.out.println("Logic1");
          System.out.println("Logic2");
          System.out.println("Logic3");
  }
}
```

**Output:**



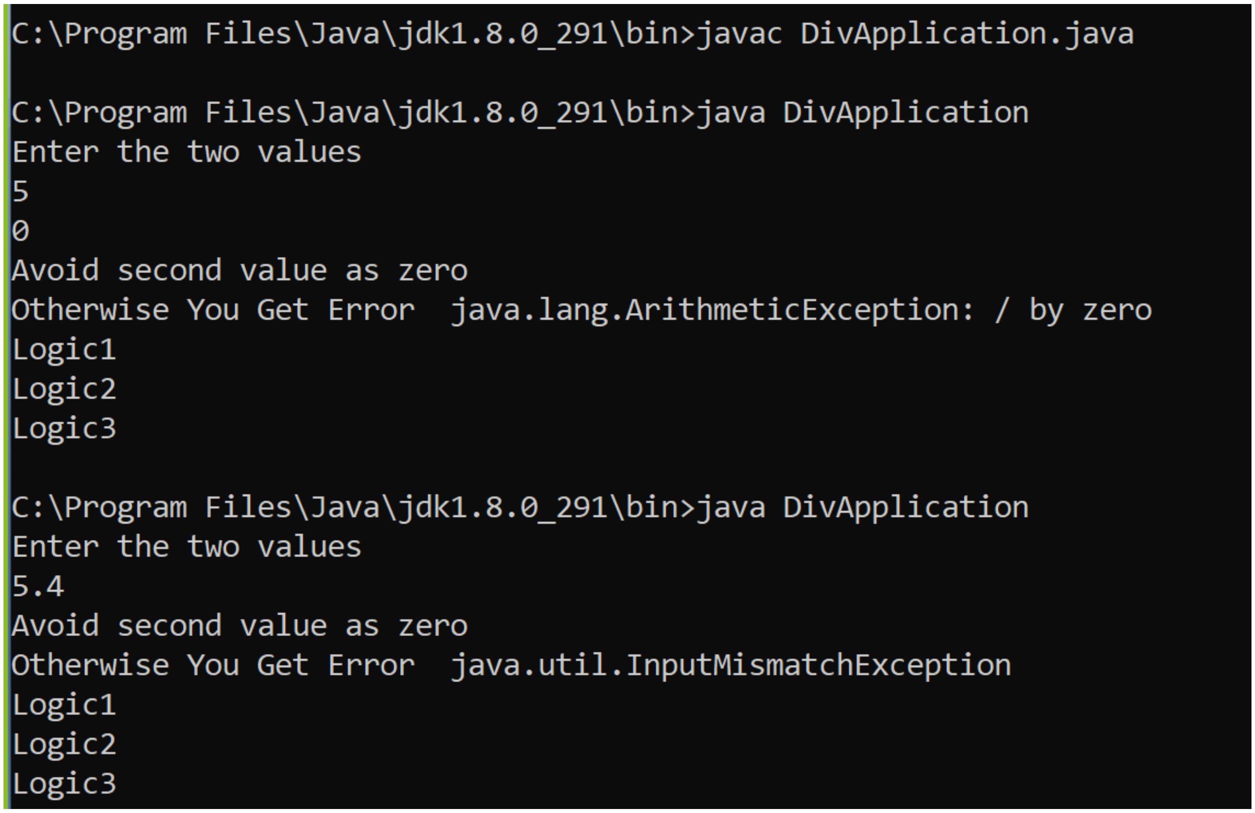
```
C:\Program Files\Java\jdk1.8.0_291\bin>javac DivApplication.java

C:\Program Files\Java\jdk1.8.0_291\bin>java DivApplication
Enter the two values
6.5
Error is  java.util.InputMismatchException
Logic1
Logic2
Logic3
```

If we write only Exception class in catch block then we can manage the any kind of Exception using single catch block.



```
import java.util.*;
public class DivApplication
{
    public static void main(String x[])
    {       Scanner xyz = new Scanner(System.in);
                int a,b,c;
      try
        {
                System.out.println("Enter the two values");
                a=xyz.nextInt();
                b=xyz.nextInt();

                c=a/b;
                System.out.printf("Division is %d\n",c);
        }
        catch(Exception ex)
        { System.out.println("Avoid second value as zero");
          System.out.println("Otherwise You Get Error  "+ex);
         }
                System.out.println("Logic1");
                System.out.println("Logic2");
                System.out.println("Logic3");
    }
}
```

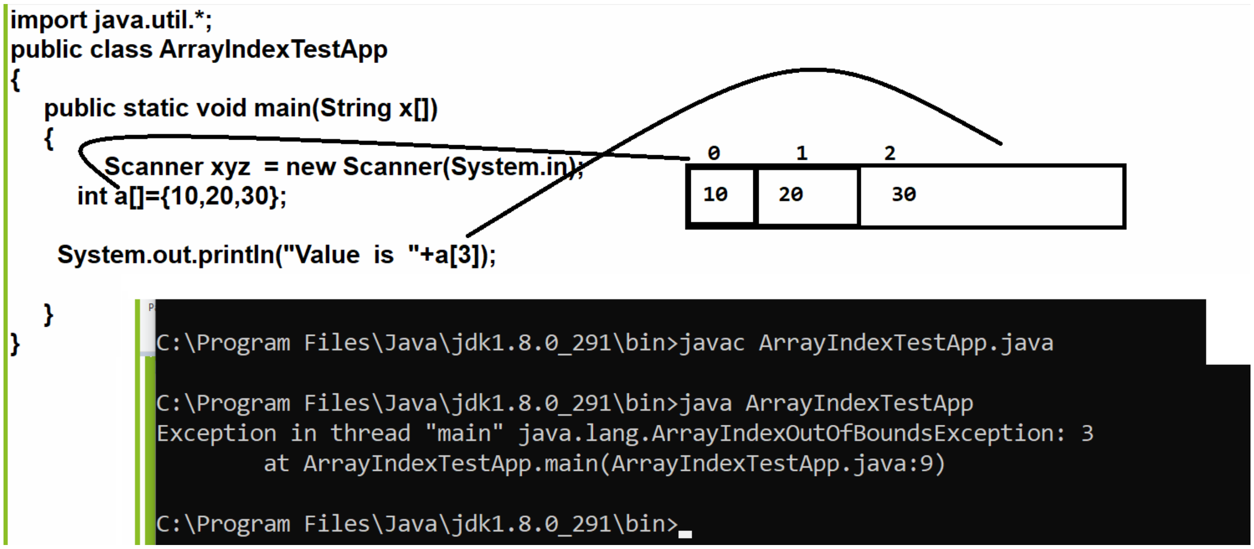
## Output



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac DivApplication.java

C:\Program Files\Java\jdk1.8.0_291\bin>java DivApplication
Enter the two values
5
0
Avoid second value as zero
Otherwise You Get Error  java.lang.ArithmeticException: / by zero
Logic1
Logic2
Logic3

C:\Program Files\Java\jdk1.8.0_291\bin>java DivApplication
Enter the two values
5.4
Avoid second value as zero
Otherwise You Get Error  java.util.InputMismatchException
Logic1
Logic2
Logic3
```

## Example for ArrayIndexOutOfBoundsException

ArrayIndexOutOfBoundsException occur when we try to store value more than index of array.



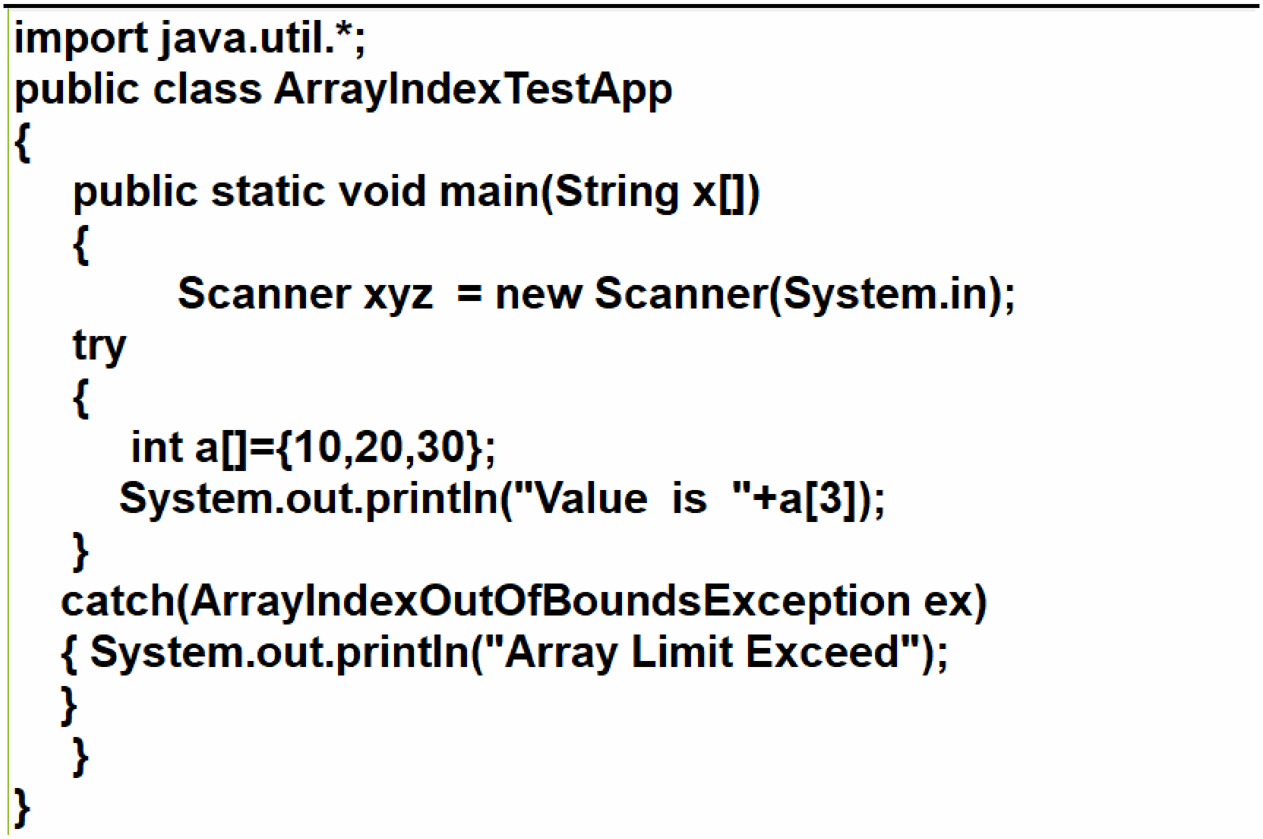
```
import java.util.*;
public class ArrayIndexTestApp
{
   public static void main(String x[])
   {
      Scanner xyz  = new Scanner(System.in);
     int a[]={10,20,30};

    System.out.println("Value  is  "+a[3]);

   }
}
```

```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ArrayIndexTestApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ArrayIndexTestApp
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3
        at ArrayIndexTestApp.main(ArrayIndexTestApp.java:9)

C:\Program Files\Java\jdk1.8.0_291\bin>
```

If we think about above code we have array with 3 elements means in array contain 0 1  and 2 index but we try to fetch element from 3rd index of array but array having last index value is 2 so JVM not found the 3rd index array so we get error ArrayIndexOutOfBoundsException

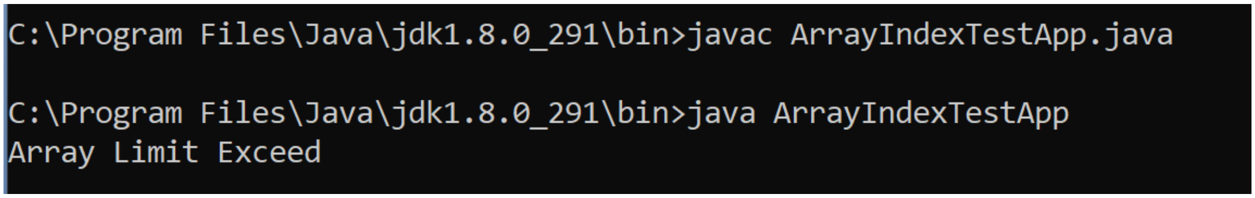
If we want to handle this problem we need to handle the ArrayIndexOutOfBoundsException in Program.

Following Example show How Handle the ArrayIndexOutOfBoundsException



```
import java.util.*;
public class ArrayIndexTestApp
{
    public static void main(String x[])
    {
          Scanner xyz  = new Scanner(System.in);
    try
    {
        int a[]={10,20,30};
       System.out.println("Value  is  "+a[3]);
    }
   catch(ArrayIndexOutOfBoundsException ex)
   { System.out.println("Array Limit Exceed");
   }
    }
}
```

**Output :**



```
C:\Program Files\Java\jdk1.8.0_291\bin>javac ArrayIndexTestApp.java

C:\Program Files\Java\jdk1.8.0_291\bin>java ArrayIndexTestApp
Array Limit Exceed
```

finally

throw

throws