1.6 EXCEPTION HANDLING:

- An exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions.
- Exceptions happen when the program encounters something it cannot handle. Common causes include:
    1. Invalid input
    2. File not found
    3. Network errors
    4. Division by zero
    5. OutOfBoundArrayIndex

There are 3 types of Exceptions in Java. They are,

1. **Checked Exception:** Caught during compile time. (e.g. IOException)
2. **Unchecked Exception:** Occurs at runtime. (e.g. NullPointerException)
3. **Error:** Serious issues. (e.g. OutOfMemoryError)

In Java, **Exception Handling** is a mechanism to:

- Detect and manage **runtime errors** (exceptions),
- Maintain **normal flow** of the program,
- Provide **graceful error recovery** instead of crashing.

There are 5 keywords in Exception Handling. They are,

- **try** : a block where we should place exception code. The try block must be followed by either catch or finally. It means, we can't use try block alone.
- **catch** : a block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone.
- **finally** : a block is used to execute the important code of the program. It is executed whether an exception is handled or not.
- **throw** : The **throw** keyword is used to throw an exception.
- **throws** : The **throws** keyword is used to declare exceptions. It doesn't throw an exception. It specifies that there may occur an exception in the method.

Common Scenarios of Java Exceptions:

- **ArithmeticException**.

  int a=50/0;
- **NullPointerException**.

  String s=null;

```
System.out.println(s.length());
```

- **NumberFormatException**.

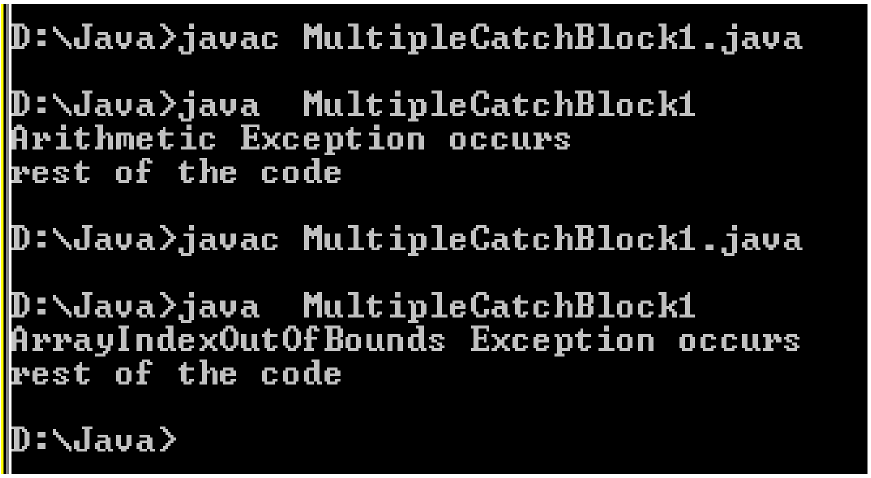
```
String s="abc";
n=Integer.parseInt(s);
```

Example Program

```
public class MultipleCatchBlock1
{
    public static void main(String[] args)
     {
       try
                {
               int a[]=new int[5];
          a[6]=30/0;
       }
       catch(ArithmeticException e)
       {
           System.out.println("Arithmetic Exception occurs");
       }

          catch(ArrayIndexOutOfBoundsException e)
          {
          System.out.println("ArrayIndexOutOfBounds Exception occurs");
          }
          catch(Exception e)
          {
          System.out.println("Parent Exception occurs");
          }
     finally()
     {
     System.out.println("rest of the code");
    }
    }
  }
```

Output:



```
D:\Java>javac MultipleCatchBlock1.java

D:\Java>java  MultipleCatchBlock1
Arithmetic Exception occurs
rest of the code

D:\Java>javac MultipleCatchBlock1.java

D:\Java>java  MultipleCatchBlock1
ArrayIndexOutOfBounds Exception occurs
rest of the code

D:\Java>
```