Exception handling:

Exception handling refers to the mechanism to handle the runtime error. The term exception is an object that is generated as the result of an unexpected error. To prevent the exceptions from crashing our program, we must write the code that detects and handles them. To handle an exception, we use a try statement.

```
try{

        (try block statements……)

}

catch( ExceptionType parameterName)

{

        (catch block statement……..)

}
```

<u>*We discussed* about following types of exception handling:</u>

1. ***Arithmetic Exception:*** Arithmetic Exception is an error that happens when an improper arithmetic condition arises.

```
public class ArithmeticExceptionDemo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

         try{
             int num =100/0;
             System.out.println("handled");
         }
         catch(ArithmeticException e){
             System.out.println("Exception Type: "+e);
         }

         System.out.println("Arithmetic Exception is handled");
    }

}
```

2. ***ArrayIndexOutOfBoundException*** : It is an error thrown to indicate that an array has been accessed with an illegal index

```
public class NullPointerExceptionDemo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
```

```
                try {
                        double[] a = new double[5];
                        a[10] = 20.21;

                }

                catch (ArrayIndexOutOfBoundsException e) {
                        System.out.println("Exception Type: " + e);
                }
                System.out.println("ArrayIndexOutOfBoundsException is handled");
        }

    }
```

3. ***NumberFormatException*** : It is an error when an attempt is made to convert a string with an incorrect format to a numeric value.

```
public class NumberFormatExceptionDemo {
    public static void main(String[] args) {
            try{
            String str= null;
            int a = Integer.parseInt(str);
            }
            catch(NumberFormatException e){
                    System.out.println("Exception Type: " + e);
            }
            System.out.println("NumberFormatException is handled");
    }

}
```

4. ***Multiple Catch Exception Handling:*** It is the method to handle the error where one try block can be followed by one or more catch blocks. Each catch block must contain a different exception handler.

```
public class MultipleCatchExhandling {

    public static void main(String[] args) {
            // TODO Auto-generated method stub
            try {
                    //int num = 100 / 0;
                    // int[] a = new int[8];
                    // a[10] = 20;
                    String str = null;
                    int b = Integer.parseInt(str);
            } catch (ArithmeticException e) {
                    System.out.println("Exception Type: " + e);
```

```
        } catch (NumberFormatException e) {
            System.out.println("Exception Type: " + e);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception Type: " + e);
        }

}

    }
```