**1. ArithmeticException:**  
We have already seen this exception in our example above. This exception occurs when we divide a number by zero. If we divide any number by zero.

int num = 25/0;//ArithmeticException

**2. NullPointerException:**  
When a variable contains null value and you are performing an operation on the variable. For example, if a string variable contains null and you are comparing with another string. Another example is when you are trying to print the length of the string that contains null.

String str = null;

//NullPointerException

System.out.println(str.length());

**3. NumberFormatException:**  
This exception occurs where there is a type mismatch. Let’s say you are trying to perform an arithmetic operator on a string variable.

String str = "beginnersbook.com";

//NumberFormatException

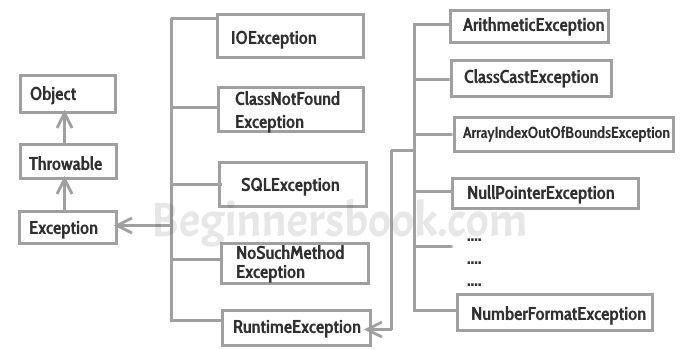
int num=Integer.parseInt(str);

**4. ArrayIndexOutOfBoundsException:**  
When you are trying to access the array index which is beyond the size of array. Here, we are trying to access the index 8 (9th element) but the size of the array is only 3. This exception occurs when you are accessing index which doesn’t exist.

int arr[]=new int[3];

//ArrayIndexOutOfBoundsException

arr[8]=100;



**Types of Exceptions**

**1) Checked exceptions**

All exceptions other than Runtime Exceptions are known as Checked exceptions as the compiler checks them during compilation to see whether the programmer has handled them or not. If these exceptions are not handled/declared in the program, you will get compilation error. For example, SQLException, IOException, ClassNotFoundException etc.

**2) Unchecked Exceptions**

Runtime Exceptions are also known as Unchecked Exceptions. These exceptions are not checked at compile-time so compiler does not check whether the programmer has handled them or not but it’s the responsibility of the programmer to handle these exceptions and provide a safe exit.

For example, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. The examples that we seen above were unchecked exceptions.  
**Note:** Compiler doesn’t enforce you to catch such exceptions or ask you to declare it in the method using throws keyword.

## Does finally block Override the values returned by try-catch block?

Yes. Finally block overrides the value returned by try and catch block