

# User defined exception in java

BY CHAITANYA SINGH | FILED UNDER: [EXCEPTION HANDLING](#)

In java we have already defined, exception classes such as `ArithmeticException`, `NullPointerException` etc. These exceptions are already set to trigger on pre-defined conditions such as when you divide a number by zero it triggers `ArithmeticException`, In the last tutorial we learnt how to throw these exceptions explicitly based on your conditions using [throw keyword](#).

In java we can create our own exception class and throw that exception using `throw` keyword. These exceptions are known as **user-defined** or **custom** exceptions. In this tutorial we will see how to create your own custom exception and throw it on a particular condition.

To understand this tutorial you should have the basic knowledge of [try-catch block](#) and [throw in java](#).

## Example of User defined exception in Java

```
/* This is my Exception class, I have named it MyException
 * you can give any name, just remember that it should
 * extend Exception class
 */
class MyException extends Exception{
    String str1;
    /* Constructor of custom exception class
     * here I am copying the message that we are passing while
     * throwing the exception to a string and then displaying
     * that string along with the message.
     */
    MyException(String str2) {
        str1=str2;
    }
    public String toString(){
        return ("MyException Occurred: "+str1) ;
    }
}

class Example1{
    public static void main(String args[]){
        try{
            System.out.println("Starting of try block");
            // I'm throwing the custom exception using throw
            throw new MyException("This is My error Message");
        }
        catch(MyException exp){
            System.out.println("Catch Block") ;
            System.out.println(exp) ;
        }
    }
}
```

```
}
```

## Output:

```
Starting of try block  
Catch Block  
MyException Occurred: This is My error Message
```

### Explanation:

You can see that while throwing custom exception I gave a string in parenthesis ( throw new MyException("This is My error Message");). That's why we have a [parameterized constructor](#) (with a String parameter) in my custom exception class.

### Notes:

1. User-defined exception must extend Exception class.
2. The exception is thrown using throw keyword.

## Another Example of Custom Exception

In this example we are throwing an exception from a method. In this case we should use throws clause in the method signature otherwise you will get compilation error saying that “unhandled exception in method”. To understand how throws clause works, refer this guide: [throws keyword in java](#).

```
class InvalidProductException extends Exception  
{  
    public InvalidProductException(String s)  
    {  
        // Call constructor of parent Exception  
        super(s);  
    }  
}  
  
public class Example1  
{  
    void productCheck(int weight) throws InvalidProductException{  
        if(weight<100){  
            throw new InvalidProductException("Product Invalid");  
        }  
    }  
  
    public static void main(String args[])  
    {  
        Example1 obj = new Example1();  
        try  
        {  
            obj.productCheck(60);  
        }  
        catch (InvalidProductException ex)  
        {  
            System.out.println("Caught the exception");  
            System.out.println(ex.getMessage());  
        }  
    }  
}
```

```
}  
}
```

### Output:

```
Caught the exception  
Product Invalid
```