



## OBJECT-ORIENTED PROGRAMMING

# EXCEPTIONS

Autores:

- Adonny Mateo Calero Arguello
- Joan Oswaldo Cobeña Zambrano
- Jordan Alexander Guaman Alcivar
- Jhordy Paul Marcillo Mora
- Lenin David Moran Palomo
- Edison Damian Verdesoto Segovia

# EXCEPTIONS IN JAVA



```
function() {
    return (
        <React.Fragment>
            <div className="py-5">
                <div className="container">
                    <Title name="our" title="p
            <div className="row">
                <ProductConsumer>
                    {(value) => {
                        console.log(value)
                    }
                </ProductConsumer>
            </div>
        </div>
    )
}
```



# DEFINITION

Exceptions in Java are unexpected events or conditions that occur during the execution of a program and interrupt its normal flow. They represent error or exceptional situations that must be detected, controlled and handled appropriately to avoid abrupt termination of the program and allow a controlled response to such events.

JAVA



Nooo! You can't just take  
something and throw it as an  
exception!

C++



```
void foo()  
{  
    // ...  
    throw [&](void**){};  
}
```

# EXCEPTION HANDLING

Exception handling allows you to control how errors and exceptional situations are responded to, preventing the program from stopping abruptly. Additionally, it provides the ability to take specific actions based on the type of exception, such as displaying error messages, performing recovery actions, or notifying the user.



## Exception

# EXAMPLES

**1.- NullPointerException:** Thrown when trying to access or use an object reference that has not been initialized, that is, when trying to access an object that has the value null.

```
String text = null;  
int length = text.length(); // Lance NullPointerException
```

**2.- ArrayIndexOutOfBoundsException:** Thrown when trying to access an invalid index on an array. The index is outside the allowed range for the size of the array.

```
int[] numbers = { 1, 2, 3 };
int value = numbers[3]; // Lance ArrayIndexOutOfBoundsException
```

**3.- FileNotFoundException:** Thrown when trying to access a file that does not exist or cannot be opened.

```
File file = new File(pathname: "archivo.txt");
Scanner scanner = new Scanner(source: file); // Lance FileNotFoundException
```

These are just a few examples of exceptions in Java. There are many more exceptions defined in the standard Java library, and it is also possible to create custom exceptions based on the needs of the program.