

# Exception Handling in Programming

Hello, welcome back. Now we're going to talk about something really important in programming: Exception handling. Now don't let that fancy term scare you. It's actually pretty simple.

— por Mayko Silva



# Understanding Exceptions

## What are Exceptions?

Let's start by thinking about what an exception is. You know how sometimes things go wrong in real life? Like maybe you try to divide a number by 0 on your calculator and it gives you an error. Well, exceptions are like that in programming. They are errors that can happen when our program is running.

## Handling Exceptions

Now let's look at some code that shows how we can handle these exceptions. I'm going to write it on the board and then we will go through it together.



# Code Breakdown: Part 1

1

## Variable Declaration

First we declare a variable called `vResult`. This is where we're going to try to store our calculation.

2

## Problematic Calculation

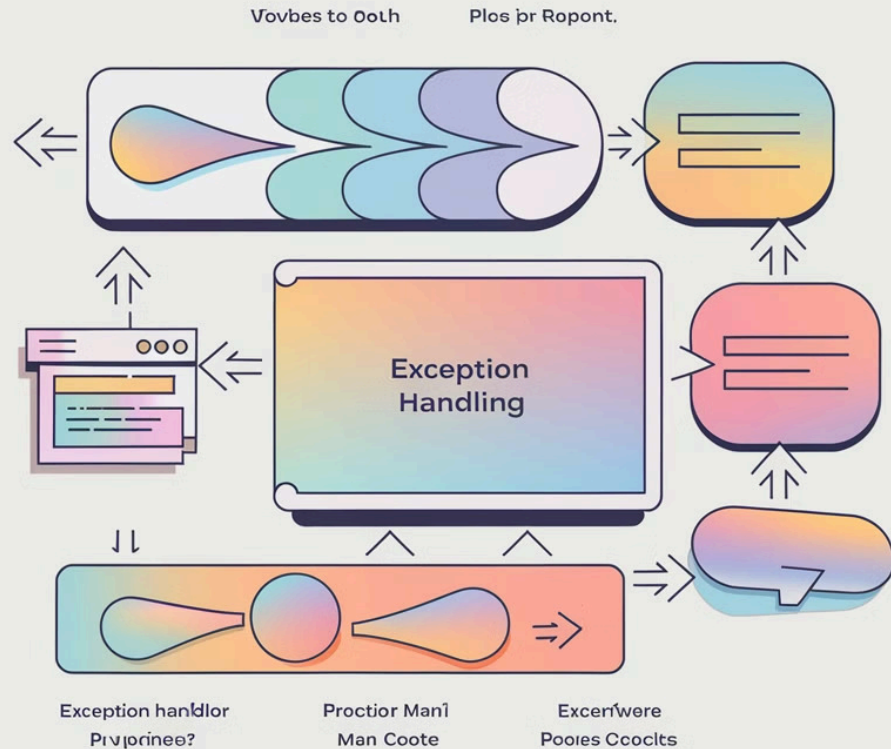
In the `begin` section we're trying to divide 10 by 0. Now we all know that it's not allowed in math, right? So this is going to cause an error.

3

## Unreachable Code

After that we have a line that tries to print the result but guess what? We are never going to get to this line because of the error we caused.

# Code Breakdown: Part 2



## Exception Handling

And then now here's where the magic happens. See that exception part? That's where we tell our program what to do when things go wrong.

## Specific Exception

We have 2 different cases here. When 0 divide then, this is specifically for when we try to divide by 0.

## General Exception

Then we have this part. When others then, this is like a catch-all for any other errors that might happen.

## Error Messages

For each of these cases, we're telling the program to print out a message explaining what went wrong.



ERROR: DIVISION BY

# Running the Program

1

Execution

So, let's see what happens when we run this program.

2

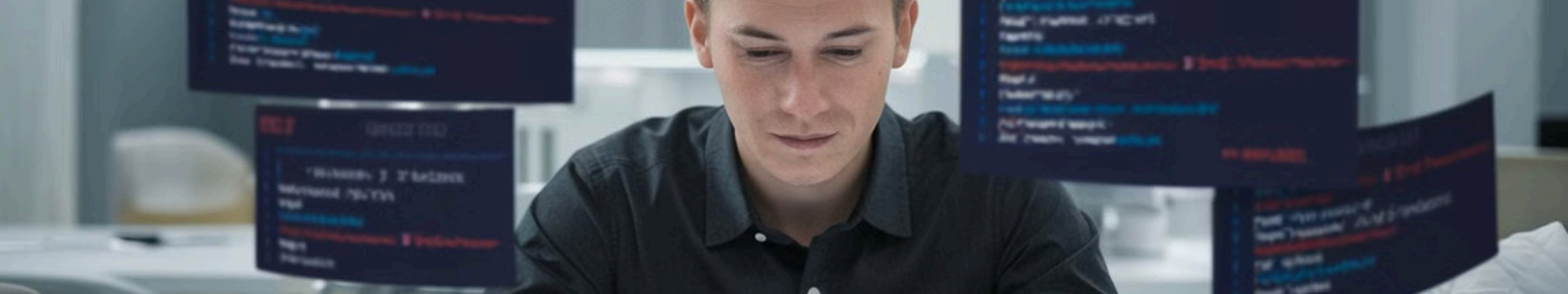
Result

As you can see, we get the message "error division by 0".

3

Benefit

So instead of crashing or giving us a confusing error message, it will nicely tell us "error division by 0". Isn't that much better?



# Importance of Exception Handling



## Smart Error Management

This is why exception handling is so important. It helps our programs deal with problems in a smart way instead of just crashing.



## Real-World Application

Now in real life, we probably wouldn't write code that divides by 0 on purpose, but unexpected things can always happen when our programs are running.



## Handling Unexpected Scenarios

Maybe a user types in the wrong kind of data, or maybe a file that we are trying to read doesn't exist. So, exception handling helps us to deal with all these potential problems.



# Examples of Exception Scenarios



## User Input Errors

When users enter unexpected data types or values.



## File Handling Issues

When trying to access non-existent files or directories.



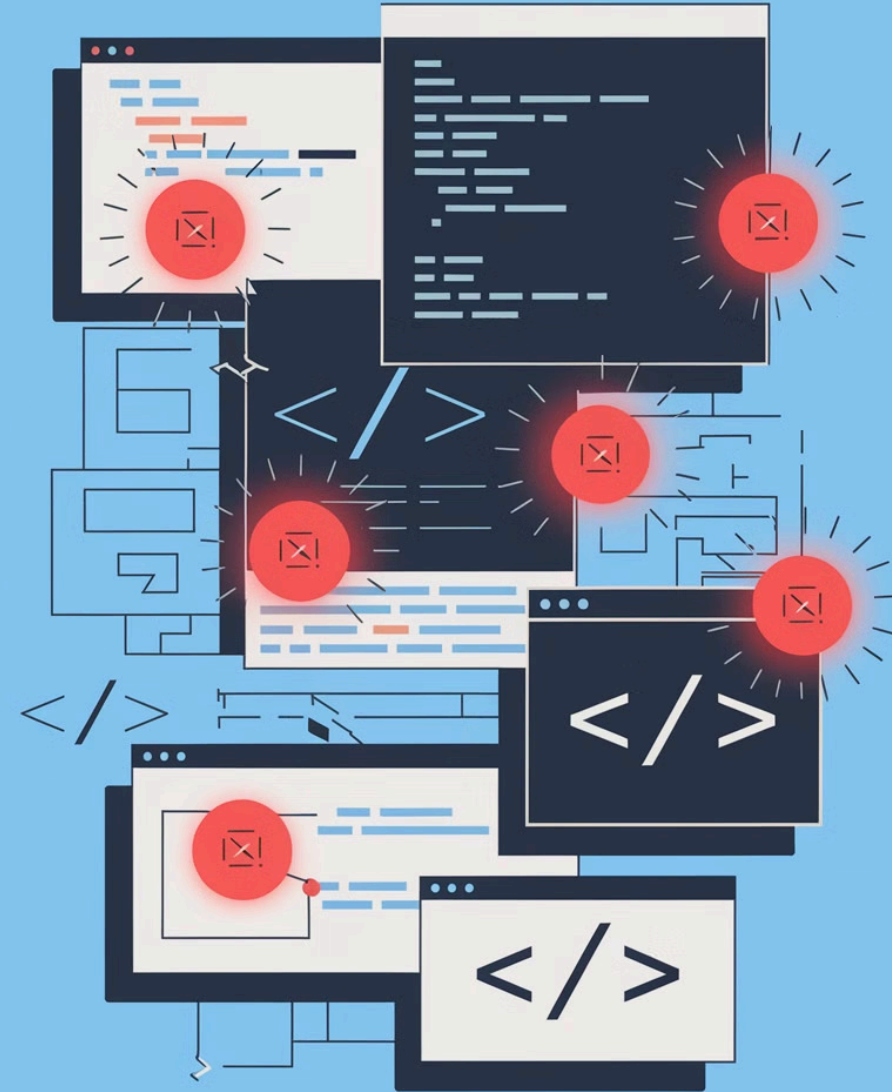
## Network Problems

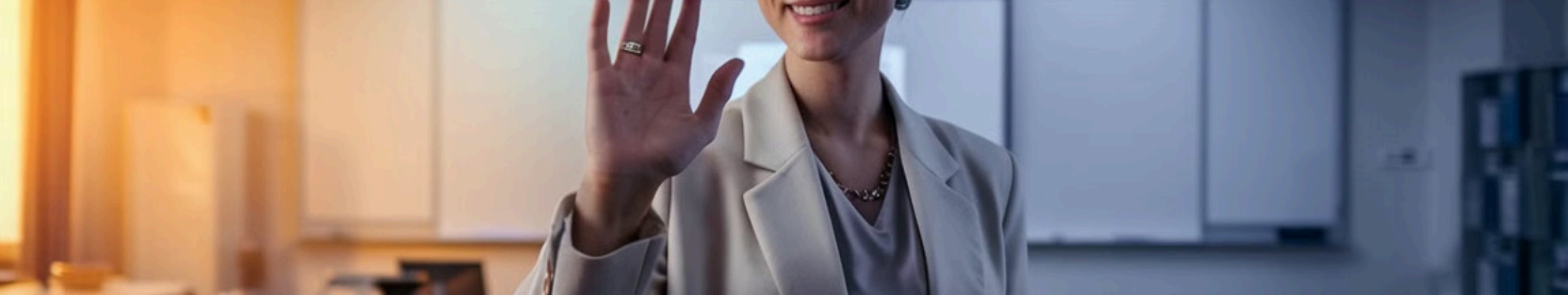
When dealing with connectivity issues or timeouts.



## Database Errors

When encountering issues with database connections or queries.





# Conclusion

Okay? See you on the next lesson!