# **🐍 Python Exception Handling – A Beginner's Guide**

## **🎯 Step 1: Why Exception Handling?**

In Python, if your code runs into an unexpected issue (like dividing by zero), it **crashes**. Exception handling helps you deal with such situations **gracefully** without stopping the whole program.

## **🧨 Step 2: What Happens Without Exception Handling?**

print("Step 1: Start")

a = 10

b = 0

result = a / b # ❌ Crash here

print("Step 2: Result is", result)

### **Output:**

Step 1: Start

ZeroDivisionError: division by zero

* The rest of the code **never runs**
* User gets a crash screen – bad experience!

## **🍽️ Step 3: Zomato App Analogy – Crash Without Handling**

print("Welcome to Zomato!")

number\_of\_items = int(input("How many items? "))

total\_price = 200 \* number\_of\_items

average\_price = total\_price / number\_of\_items # ❌ Division by zero

print("Average price per item:", average\_price)

### **Input: 0**

Output: ZeroDivisionError

## **🛡️ Step 4: Handling It Using try-except**

try:

number\_of\_items = int(input("How many items? "))

total\_price = 200 \* number\_of\_items

average\_price = total\_price / number\_of\_items

print("Average price:", average\_price)

except ZeroDivisionError:

print("❌ You cannot order 0 items.")

except ValueError:

print("❌ Please enter a valid number.")

finally:

print("✅ Thank you for using Zomato!")

## **🧠 Step 5: if-else vs try-except**

### **Q: “Why not just use if-else?”**

# if-else only checks known problems

if number\_of\_items == 0:

print("❌ Cannot divide by zero")

else:

print(200 / number\_of\_items)

❌ Crashes if user enters "abc" → ValueError  
 ✅ try-except can handle **unexpected errors** too.

## **📋 Step 6: Common Python Exceptions**

| **Exception Name** | **When It Happens** |
| --- | --- |
| ZeroDivisionError | Divide by 0 |
| ValueError | Invalid conversion (int("abc")) |
| TypeError | Mixing data types ("a" + 1) |
| IndexError | List index out of range |
| KeyError | Dictionary key not found |
| FileNotFoundError | File doesn’t exist |
| AttributeError | Method not found on object |

## **🔍 Step 7: Catch All Exceptions (If You're Not Sure)**

try:

# risky code

except Exception as e:

print("⚠️ Error occurred:", e)

✅ Safest for unknown cases  
 ❌ Not ideal for production if overused

## **❌ Step 8: What Not to Do**

try:

risky\_code()

except:

pass # ❌ BAD PRACTICE – hides all bugs

Avoid silent fails. Always log or handle clearly.

## **✅ Step 9: Recap Summary**

| **Concept** | **Use Case Example** |
| --- | --- |
| try block | Code that might fail |
| except ErrorType | Handle specific error |
| except Exception as e | Handle unknown error |
| else block | Runs if no exception occurred |
| finally block | Always runs (cleanups, messages) |

## **🧪 Step 10: Practice Exercise**

Write a program that:

* Asks for 2 numbers
* Divides them
* Handles ZeroDivisionError and ValueError
* Always prints "Thanks for using the calculator"

## **💡 PROJECT IDEA: “Zomato Order Calculator with Exception Handling”**

### **📝 1. Project Description (for Portfolio / GitHub / Resume)**

#### **📌 Project Title:**

**Zomato Order Calculator (Exception Handling Demonstration)**

#### **🛠 Description:**

A Python-based console application that simulates a Zomato-like food order system. The project demonstrates real-time exception handling by managing invalid user inputs, such as entering zero quantity, non-numeric values, and handling division errors. It uses Python’s try-except-else-finally blocks to gracefully catch and handle runtime errors, enhancing user experience and application stability.

#### **🎯 Objective:**

To demonstrate how to handle real-world edge cases and exceptions in user inputs using Python's built-in error-handling constructs.

#### **✅ Features:**

* Input handling for number of items
* Error messages for zero or invalid input
* Always shows a friendly thank-you message (via finally)
* Covers both specific and generic exceptions
* Clean code with comments

#### **🧰 Tech Stack:**

* Language: Python 3
* IDE: VS Code / PyCharm / Jupyter

#### **📂 File Structure:**

zomato\_calculator.py

README.md

#### **📦 How to Run:**

python zomato\_calculator.py

#### **🧪 Sample Input/Output:**

How many items? 0

❌ You cannot order 0 items.

✅ Thank you for using Zomato!

#### **📌 Concepts Used:**

* try-except-else-finally
* ZeroDivisionError
* ValueError
* Clean code practices

## **🎤 2. How to Present in an Interview**

### **🗣️ When They Ask:**

**"Can you explain a Python project you've worked on?"**

### **✅ Suggested Answer (Script Style):**

*"Sure! I built a mini Zomato Order Calculator in Python. The idea was to simulate a food-ordering scenario where users input the number of items they want to order. I used exception handling to gracefully manage unexpected inputs — like dividing by zero when the user enters 0, or catching ValueError when they type text instead of a number. I also included a finally block to always thank the user, showing how even cleanups can be done safely. This helped me master how real-world apps use exception handling to prevent crashes."*

*"I designed the code to be clean and readable and even added logging in the final version to simulate production-level practices."*

### **🔥 Bonus Points You Can Add:**

* "Handled unknown exceptions using except Exception as e: to make the app crash-proof."
* "This experience made me realize how important it is to anticipate user behavior and build fault-tolerant systems."
* "I later extended it to include a menu system and price selection using dictionaries and loops."

## **🧠 Tip: Show It, Don't Just Tell**

* Upload your .py file to GitHub ✅
* Add screenshots or sample input/output in README.md 📸
* Paste the link in your resume or LinkedIn portfolio 🌐

### **About the Author**

**Gowtham SB** is a **Data Engineering expert, educator,** **and content creator** with a passion for **big data technologies, as well as cloud and Gen AI** . With years of experience in the field, he has worked extensively with **cloud platforms, distributed systems, and data pipelines**, helping professionals and aspiring engineers master the art of data engineering.

Beyond his technical expertise, Gowtham is a **renowned mentor and speaker**, sharing his insights through engaging content on **YouTube and LinkedIn**. He has built one of the **largest Tamil Data Engineering communities**, guiding thousands of learners to excel in their careers.

Through his deep industry knowledge and hands-on approach, Gowtham continues to **bridge the gap between learning and real-world implementation**, empowering individuals to build **scalable, high-performance data solutions**.

𝐒𝐨𝐜𝐢𝐚𝐥𝐬

🎥𝐘𝐨𝐮𝐓𝐮𝐛𝐞 - https://www.youtube.com/@dataengineeringvideos

📸𝐈𝐧𝐬𝐭𝐚𝐠𝐫𝐚𝐦 - <https://instagram.com/dataengineeringtamil>

📸𝐈𝐧𝐬𝐭𝐚𝐠𝐫𝐚𝐦 - [https://instagram.com/](https://instagram.com/dataengineeringtamil)thedatatech.in

🤝𝐂𝐨𝐧𝐧𝐞𝐜𝐭 𝐟𝐨𝐫 𝟏:𝟏 - https://topmate.io/dataengineering/

💼𝐋𝐢𝐧𝐤𝐞𝐝𝐈𝐧 - https://www.linkedin.com/in/sbgowtham/

🌐𝐖𝐞𝐛𝐬𝐢𝐭𝐞 - https://codewithgowtham.blogspot.com

💻𝐆𝐢𝐭𝐇𝐮𝐛 - http://github.com/Gowthamdataengineer

💬𝐖𝐡𝐚𝐭𝐬 𝐀𝐩𝐩 - https://lnkd.in/g5JrHw8q

📧𝐄𝐦𝐚𝐢𝐥 - atozknowledge.com@gmail.com

📱𝐀𝐥𝐥 𝐌𝐲 𝐒𝐨𝐜𝐢𝐚𝐥𝐬 - <https://lnkd.in/gf8k3aCH>