



Python Testing ZGuide: Manual Test vs Test Case vs Exception Handling

1 WHY TEST YOUR CODE?

To avoid:

- Bugs in production
 - App crashes
 - Wrong output
 - Poor user experience
-

2 ✖️ Manual Testing vs Automated Test Cases

Feature	Manual Testing	Test Case (Automated Testing)
How it works	Run code manually and check output	Write <code>assert</code> statements to auto-check
Fast for small code	<input checked="" type="checkbox"/> Yes	⚠️ Overhead at first, but pays off later
Repeatable	✗ You must test again manually	<input checked="" type="checkbox"/> Run all tests with <code>pytest/unittest</code>
Catch regressions	✗ Hard to track	<input checked="" type="checkbox"/> Alerts when old bug comes back
Example	<code>print(divide(10, 2))</code>	<code>assert divide(10, 2) == 5</code>

3 🛡 Exception Handling vs Test Case

Feature	Exception Handling	Test Case (unittest / pytest)
Purpose	Avoid crashes	Validate that logic is correct
When used	In production app	During development
Handles invalid input?	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes (you can test invalid input too)
Detects wrong logic?	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> Yes
Example	<code>try: a/b except:</code>	<code>assert divide(6, 2) == 3</code>

4 ✅ Sample Code + Manual Testing

```
def divide(a, b):
    return a / b

print(divide(10, 2)) # ✅ OK
print(divide(10, 0)) # ❌ ZeroDivisionError (crash)
```

5 ✅ Improved Code With Exception Handling

```
def divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        return "Cannot divide by zero ❌"
```

Manual test:

```
print(divide(10, 0)) # ✅ "Cannot divide by zero ❌"
```



Add Test Cases with unittest

```
import unittest

def divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        return "Cannot divide by zero ✗"

class TestDivideFunction(unittest.TestCase):
    def test_divide_normal(self):
        self.assertEqual(divide(10, 2), 5.0)

    def test_divide_zero(self):
        self.assertEqual(divide(10, 0), "Cannot divide by zero ✗")

if __name__ == '__main__':
    unittest.main()
```

Run with:

```
python test_divide.py
```



Same with pytest (Simpler)

```
# test_divide.py
from your_module import divide

def test_divide_normal():
    assert divide(10, 2) == 5.0

def test_divide_zero():
    assert divide(10, 0) == "Cannot divide by zero ✗"
```

Run with:

pytest test_divide.py



🔥 What Happens if You Introduce a Bug?

If you change this:

```
return a * b # ❌ by mistake
```

You will still get:

```
FAILED test_divide_normal - assert 20 == 5.0
```

⌚ So your test case **instantly catches bugs**.

⬅ END Final Summary

Concept	What It Helps With
Manual Testing	Quick checks during development
Exception Handling	Graceful crash protection in real time
Unit Test / Pytest	Automated correctness check, regression safety

Gowtham SB

www.linkedin.com/in/sbgowtham/

Instagram - @dataengineeringtamil

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/thedatatech.in>

 **Connect for 1:1** - <https://topmate.io/dataengineering/>

 **LinkedIn** - <https://www.linkedin.com/in/sbgowtham/>

Gowtham SB

www.linkedin.com/in/sbgowtham/

Instagram - @dataengineeringtamil

 **Website** - <https://codewithgowtham.blogspot.com>

 **GitHub** - <http://github.com/Gowthamdataengineer>

 **WhatsApp** - <https://lnkd.in/g5JrHw8q>

 **Email** - atozknowledge.com@gmail.com

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