

Lab Assignment & Solution



Cybersecurity Professional Program
Introduction to Python
for Security

File System & Error Handling

PY-04-LS1

Try & Except Practice

Copyright © 1996–2021 HackerU Ltd.
All Rights Reserved.

Note: Solutions for the instructor are shown inside the green box.



Lab Objective

Understand error handling and implementation of solutions for expected system errors.



Lab Mission

Use ***try*** and ***except*** to handle code errors.



Lab Duration

10–20 minutes



Requirements

- Basic knowledge of Python
- Basic knowledge of error handling



Resources

- Environment & Tools
 - Windows, Linux, MacOS
 - PyCharm
 - Python 3



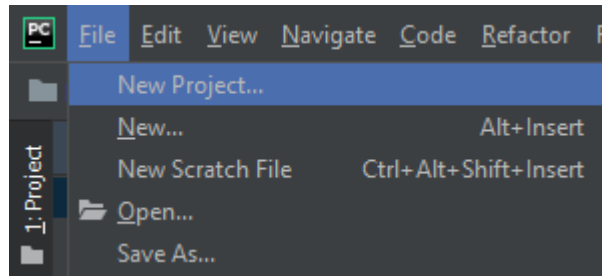
Textbook References

- Chapter 4: File System and Error Handling
 - Section 1: Error Handling

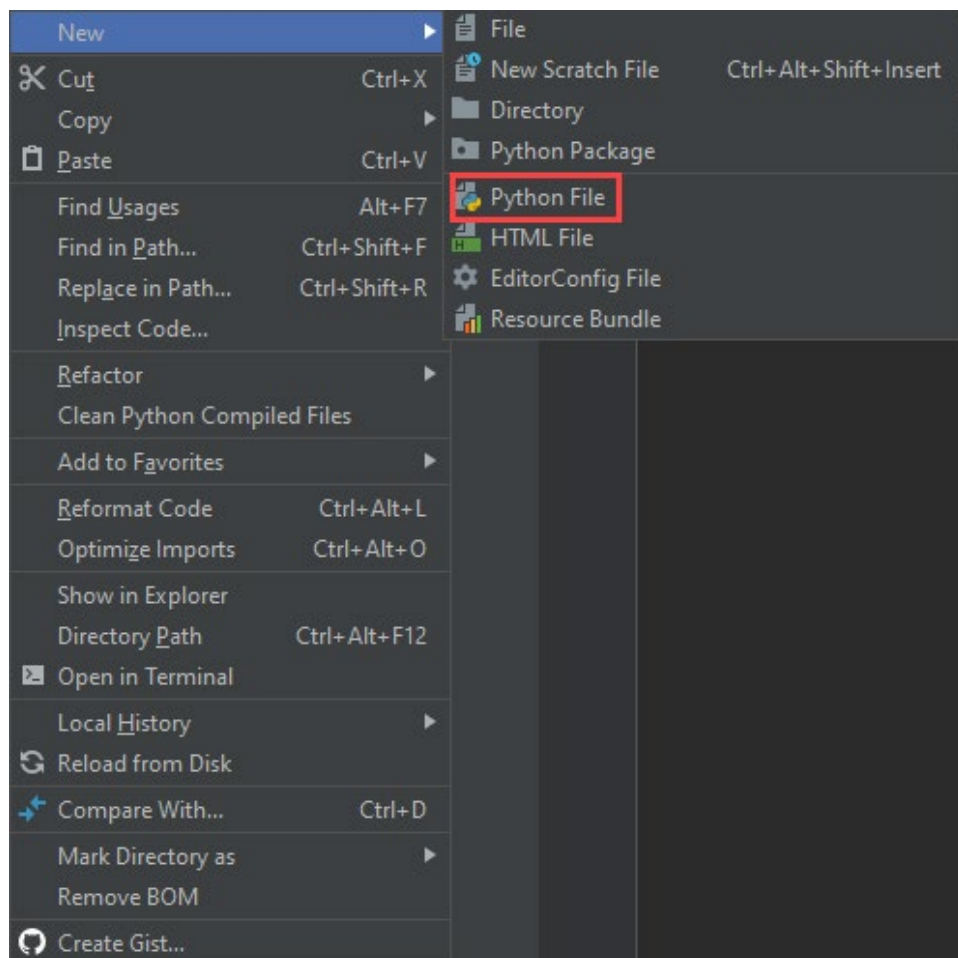
Lab Task: Division by Zero

Create a program that receives a number from the user and divides it by zero. As this operation is invalid, the program must handle the error accordingly.

- 1 Open PyCharm, click **File** at the top left, and select **New Project...**



- 2 Create a new Python file in PyCharm by right-clicking the project you created and selecting **New > Python File**.



- 3 Request a number from the user and assign it to a variable.

```
num1 = int(input("Please enter a number: "))
```

- 4 Create a new variable with the value 0.

```
num1 = int(input("Please enter a number: "))  
num2 = 0
```

- 5 Divide the first variable by the second variable and print the result.
As these operations need to be handled appropriately, begin the code with a **try** error-handling block.

```
try:  
    num1 = int(input("Please enter a number: "))  
    num2 = 0  
    div = num1/num2  
    print(div)
```

- 6 Write an **except** block to catch the **ZeroDivisionError** exception.

```
try:  
    num1 = int(input("Please enter a number: "))  
    num2 = 0  
    div = num1/num2  
    print(div)  
except ZeroDivisionError:  
    print("Can't calculate it")
```

- 7 Create another exception using the built-in ***TypeError***. Run the code and insert a word instead of a number. Note how the ***ZeroDivisionError*** exception is not executed. Why?

```
try:
    num1 = int(input("Please enter a number: "))
    num2 = 0
    div = num1/num2
    print(div)
except ZeroDivisionError:
    print("Can't calculate it")
except ValueError:
    print("Something went wrong!")
```

The first exception responds because there is no division by zero. This in turn triggers the ***ZeroDivisionError***.