Exception Handling

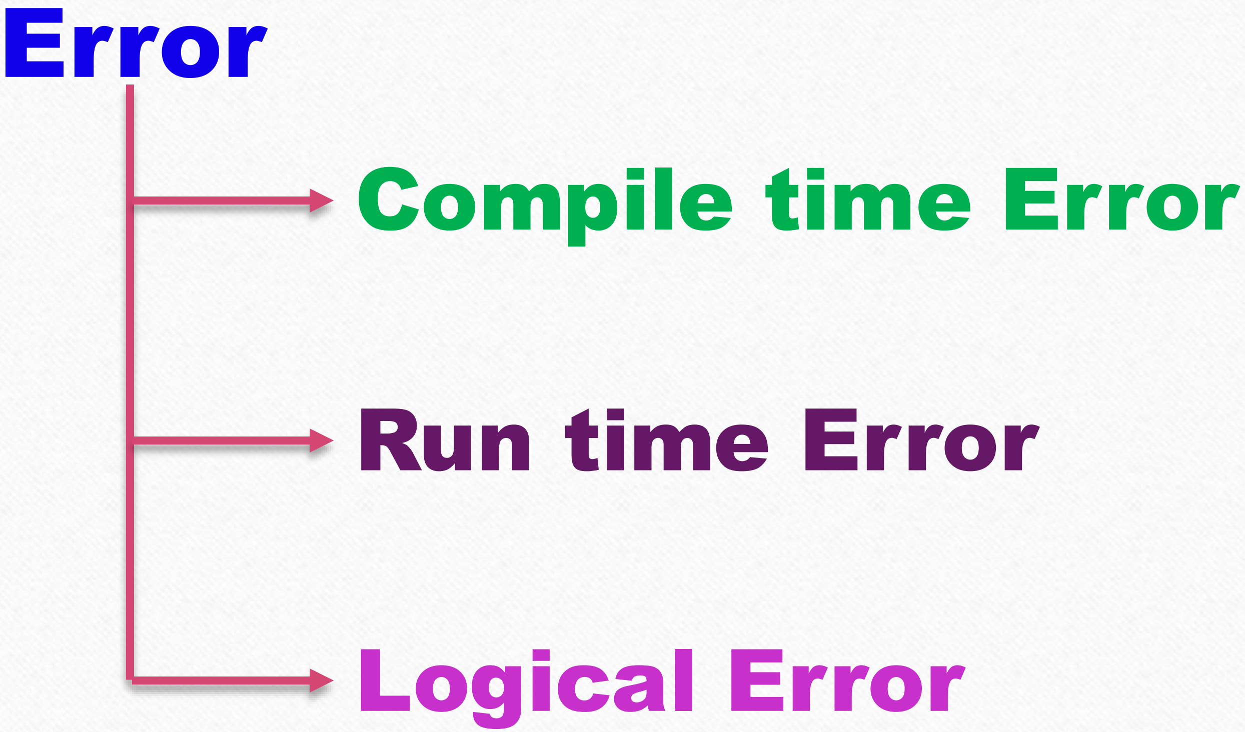
Mohammad Tasin (Tasin Coder)

Agenda

- What is Error
- Introduction to Exceptions
- Handling Exceptions
- Default Exceptions
- use of else in exception handling
- raising exceptions

What is Error

- Also known as bugs, are mistakes or faults in a program that cause it to behave unexpectedly.



Error

Compile time Error

Run time Error

Logical Error

- **Compile time Error → Compile time errors are the most common errors that occur due to typing mistakes or not following the proper syntax of the programming language**
- **Logical Error → Logical errors are bugs that cause a program to operate incorrectly but not to terminate abnormally. They produce undesired output or other behavior, although they may not immediately be recognized as such.**

Introduction to Exceptions

- Errors which occur during program execution (run-time) after successful compilation are called run-time errors.
- One of the most common run-time error is division by zero also known as Division error.

1. Normal Statement
2. Critical Statement

Handling Exceptions

- **try** **is a keyword**
- **except** **is a keyword**
- **finally** **is a keyword**

```
a = 10
try:
    b = int(input("Enter a number : "))
    c = a/b
except ValueError:
    print("Value Error")
finally:
    print(c)
```

Default Exceptions

- **Default exception miens handle any types of exception, but you don’t know in this situation which exceptions use Exception method.**

```
a = 10
c = 0
try:
    b = int(input("Enter a number : "))
    c = a/b
except Exception as e:
    print(e)
print(c)
```

use of else in exception handling

- If any exception is found in the try block then the else block code will not run.

```
a = 10
try:
    b = int(input("Enter a number : "))
    c = a/b
except ValueError:
    print("Value Error")
except ZeroDivisionError:
    print("Division By Zero Error")
else:
    print(c)
```

raising exceptions

- Create a exception in the block use raise keyword

```
a = 10
c = 0
try:
    b = int(input("Enter a number : "))
    if b == 5:
        raise ValueError("Value is String")
    c = a/b
except ValueError:
    print("Value Error")
print(c)
```