Dhirubhai Ambani University

(Formerly known as DA-IICT)

Topic: Error and Exceptions

Course: Programming Lab

Course Code- PC503

Dr. Ankit Vijayvargiya

Assistant Professor

Room No. 4205, Faculty Block 4

Email: ankit_Vijayvargiya[at]dau.ac.in

Phone: 079-68261628(O), 7877709590(M)

Dhirubhai Ambani University
Formerly known as DA-IICT

Errors and Exceptions

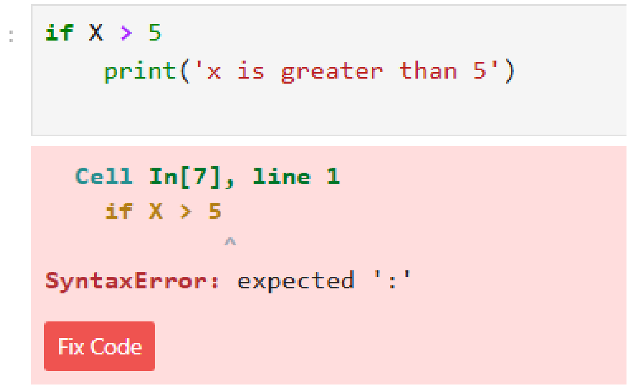
An error is a problem in the program that prevents it from running as expected.

These are two main types:

1. Syntax Errors (Compile-time error)

Example:

```
if X > 5
    print("x is greater than 5")
```



```
if X > 5
    print('x is greater than 5')
```

```
  Cell In[7], line 1
    if X > 5
            ^
SyntaxError: expected ':'
```

2. Logical Errors (Runtime error)

i) This occur while program is running.

ii) They are technically exceptions

Exception: An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of program.

- Zero Division Error
- Type Error
- Name Error
- Attribute Error
- Index Error
- Key Error
- I/O Error

Errors and Exceptions

Four keywords:

try **except** **else** **finally**

try: In which those lines or codes are placed which might contain errors. So, we have to handle because our program will not close abruptly.

except: This is the block in which statements handle the exceptions are written.

else: The statements inside the else block are executed when there is no exception found in try block.

finally: The statement(s) inside finally block are executed irrespective of wheather any exception is found in try block or not.

Dhirubhai Ambani University
Formerly known as DA-IICT

Errors and Exceptions

Some Imp points:

- A single try statements can have multiple except statements. This is useful when the try block contains statements that throw different types of exceptions.

```
try:
        “body of try”
except some_exception:
        “body of except”
except some_another_exception:
        “body of except”
```

- You can provide a generic except clause which handles any exceptions.

```
try:
        a=10/0
except:
        print(“An error occurred”)
```

- You can have only at most one else block and at most one finally block

```
try:
        “body of try”
except some_exception:
        “body of except”
except some_another_exception:
        “body of except”
else:
        “body of else”
finally:
        “body of finally
```

Dhirubhai Ambani University
Formerly known as DA-IICT

Errors and Exceptions

Exception handling is a way to manage errors that might happen during the program, instead of crashing the whole program.

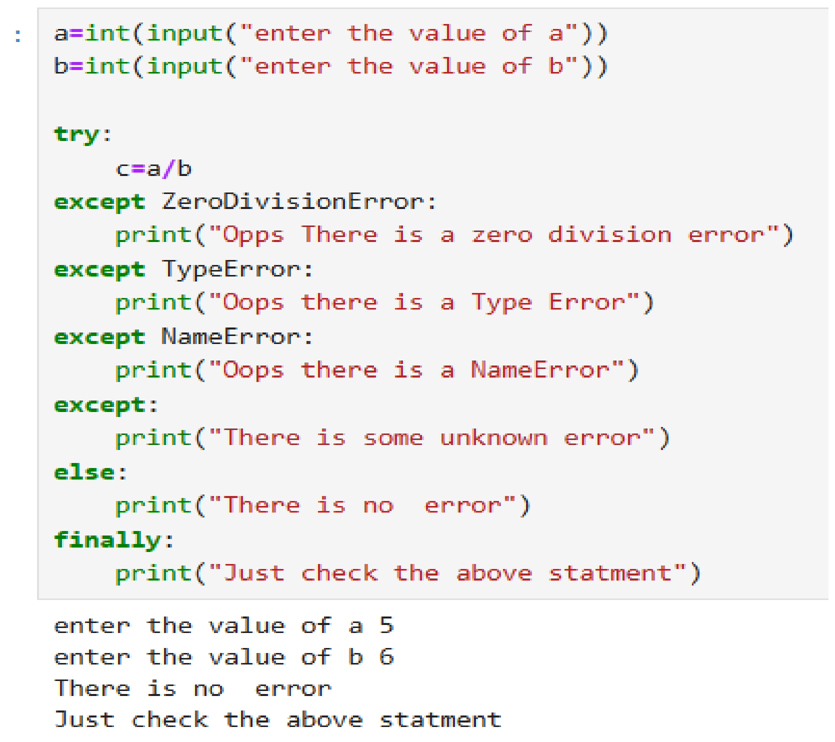
There are two ways:

1. Built-in exception
   - Pre-defined or already available in python

2. User defined exception
   - You can define your own exception

Errors and Exceptions

Built-in-exception

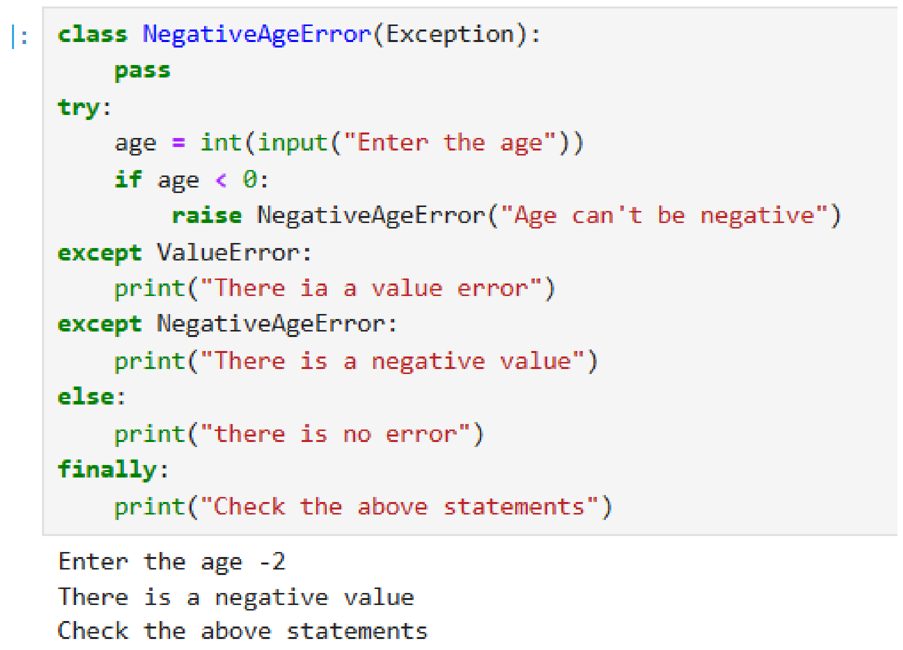


```
:  a=int(input("enter the value of a"))
   b=int(input("enter the value of b"))

   try:
       c=a/b
   except ZeroDivisionError:
       print("Opps There is a zero division error")
   except TypeError:
       print("Oops there is a Type Error")
   except NameError:
       print("Oops there is a NameError")
   except:
       print("There is some unknown error")
   else:
       print("There is no  error")
   finally:
       print("Just check the above statment")
```

```
enter the value of a 5
enter the value of b 6
There is no  error
Just check the above statment
```

User-defined-exception



```
]: class NegativeAgeError(Exception):
       pass
   try:
       age = int(input("Enter the age"))
       if age < 0:
           raise NegativeAgeError("Age can't be negative")
   except ValueError:
       print("There ia a value error")
   except NegativeAgeError:
       print("There is a negative value")
   else:
       print("there is no error")
   finally:
       print("Check the above statements")
```

```
Enter the age -2
There is a negative value
Check the above statements
```

Dhirubhai Ambani University
Formerly known as DA-IICT

Exercise

- **Create a list of dictionaries representing 5 employees in the company. Each employee's dictionary should contain:**
  - ID: Employee ID (integer)
  - name: Name of the employee (string)
  - department: Department the employee belongs to (string)
  - salary: Salary of the employee (float)

- **add a new employee to the list**

- **remove an employee by their ID**

- **update the salary of an employee based on their ID**

- **find an employee by their ID and print their details**

Ensure that when adding or updating employees:

- IDs are unique integers
- Names and departments are non-empty strings
- Salary is a positive float
  Raise appropriate custom exceptions if validation fails like:
  - EmployeeNotFoundError
  - DuplicateEmployeeIDError
  - InvalidEmployeeDataError.

Dhirubhai Ambani University
Formerly known as DA-IICT