Defensive Programming - Error Handeling

Hemant Thapa

HT22100004510

Type of Error

1.SyntaxError

Compilation error refers to a state when a compiler fails to compile a piece of computer program source code, either due to errors in the code, or, more unusually, due to errors in the compiler itself. A compilation error message often helps programmers debugging the source code.

1. A syntax error is an error in the syntax of a sequence of characters.
2. A syntax error occurs due to fault in the program syntax
3. It is easier to identify syntax error.
4. The compile indicates the syntax error with the location and what the error is.

Common python syntax error inculdes:

1. Leaving out a keyword or putting it in the wrong place
2. Misspelling a keyword (for example a function or variable name)
3. Leaving out an important symbol(such as a colon, comma or parentheses)
4. Incorrect indentation
5. Leaving an empty block(for example, an if statement containing no intented statements)

In [7]:
```
1 user_name = str(input("Enter your user name: ") #missing parentheses
2 if user_name == "harry" #missing column
3     print("\nyou're welcome", user_name)
```

```
  Input In [7]
    print("\nyou're welcome", user_name)
    ^
SyntaxError: invalid syntax
```

In [128]:
```
1 user_name = str(input("Enter your user name: ")) #fixed - missing parentheses
2 if user_name == "harry": #fixed - missing column
3     print("\nyou're welcome", user_name)
```

```
Enter your user name: harry

you're welcome harry
```

In [119]:
```
1 #The reason for this error is that forgot a closing double quote at the end of the string.
2 #String literals can be enclosed in matching single quotes (') or double quotes (").
3 def printMsg():
4     return "This is a test #Missing quotes
5 printMsg()
```

```
  Input In [119]
    return "This is a test #Missing quotes
                                          ^
SyntaxError: EOL while scanning string literal
```

In [118]:
```
1 def printMsg():
2     return "This is a test"
3 printMsg()
```

Out[118]:
```
'This is a test'
```

2. Indentation error

In [5]:
```
1 user_name = str(input("Enter your user name: "))
2 if user_name == "harry":
3 print("\nyou're welcome", user_name) #incorrect indentation
```

```
  Input In [5]
    print("\nyou're welcome", user_name)
    ^
IndentationError: expected an indented block
```

In [127]:
```
1 user_name = str(input("Enter your user name: "))
2 if user_name == "harry":
3     print("\nyou're welcome", user_name) #correct indentation
```

```
Enter your user name: harry

you're welcome harry
```

3. name error

In [10]:
```
1 user_name = str(input("Enter your user name: "))
2 if user_name == "harry":
3      print("\nyou're welcome", user_name)
```

```
Enter your user name: asda

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [10], in <cell line: 2>()
      1 user_name = str(input("Enter your user name: "))
----> 2 if user == "harry":
      3     print("\nyou're welcome", user_name)

NameError: name 'user' is not defined
```

Runtime Error

1. Type Error

In [125]:
```
#TypeError: '>=' not supported between instances of 'str' and 'int'
#If you try to compare a string and an integer, you'll encounter an error that says "not supported between instances
print("\nWelcome to Bamboo")
#input need to be str or int
user_age = input("How old are you? ")

if user_age>=18: #"TypeError: '<' not supported between instances of 'str' and 'int'" occurs when we use a compariso
    print("\nYou can old enough to enter")
else:
    print("\nSorry, you aren't old enough")

```

```
Welcome to Bamboo
How old are you? 23

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [125], in <cell line: 7>()
      4 #input need to be str or int
      5 user_age = input("How old are you? ")
----> 7 if user_age>=18: #"TypeError: '<' not supported between instances of 'str' and 'int'" occurs when we use a comp
arison operator between values of type str and int.
      8     print("\nYou can old enough to enter")
      9 else:

TypeError: '>=' not supported between instances of 'str' and 'int'
```

In [124]:

```
#TypeError: '>=' not supported between instances of 'str' and 'int'

print("Welcome to Bamboo\n")
#using int() to specify user input is integer
user_age = int(input("How old are you? "))

if user_age>=18:
    print("\nYou can old enough to enter")
else:
    print("\nSorry, you aren't old enough")
```

```
Welcome to Bamboo

How old are you? 23

You can old enough to enter
```

2. Index Error

In [121]:

```
#IndexError: list index out of range
#"List index out of range" error occurs in Python when we try to access an undefined element from the list.
print("Shopping list\n")
shopping_list = ["Coat", "Shirt", "Shoes", "Watch", "Tie"]
print(shopping_list[5]) #incorrect indexing
```

```
Shopping list


---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [121], in <cell line: 5>()
      3 print("Shopping list\n")
      4 shopping_list = ["Coat", "Shirt", "Shoes", "Watch", "Tie"]
----> 5 print(shopping_list[5])

IndexError: list index out of range
```

In [122]:
```
#Fixing - IndexError: list index out of range
print("Shopping list\n")
shopping_list = ["Coat", "Shirt", "Shoes", "Watch", "Tie"]
print(shopping_list[4]) #correct indexing

```

```
Shopping list

Tie
```

3. Zero Division Error

In [22]:
```
alpha = 0
beta = 1
gamma = beta/alpha
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Input In [22], in <cell line: 3>()
      1 alpha = 0
      2 beta = 1
----> 3 gamma = beta/alpha

ZeroDivisionError: division by zero
```

In [129]:

```python
#ZeroDivisionError: division by zero
#"ZeroDivisionError: float division by zero" occurs when we try to divide a floating-point number by 0.
# To solve the error, use an if statement to check if the number you are dividing by is not zero, or handle the erro
alpha = 0
beta = 20
try:
    gamma = beta/alpha
except ZeroDivisionError:
    gamma = 0

print(gamma)  #0
```

```
0
```

3. Logical Error

1. A logical error is an error in a program that causes it to operate incorrectly
2. A logical error occurs due to a fault in the algorithms
3. It is comparatively difficult to identify a logical error.
4. The programmer has to detect the error by himself.

Logical errors, also known as semantic errors, cause the programme to behave incorrectly but do not usually cause the programme to crash. A programme with logic errors, unlike one with syntax errors, can be run, but it does not function as intended.

1. Example (Average of number)

In [100]:
```
num1 = float(input('Enter a number: '))
num2 = float(input('Enter a number: '))
num3 = float(input('Enter a number: '))

#the order of operations in arithmetic (the division is evaluated before addition)
#the program will not give the correct answer
average = num1+num2+num3/3
print('\nThe average of the numbers is:',average,"\n")
print(average, "Average is incorrect due to logical error, (the division is evaluated before addition) ")
```

```
Enter a number: 6
Enter a number: 6
Enter a number: 6

The average of the numbers is: 14.0

14.0 Average is incorrect due to logical error, (the division is evaluated before addition)
```

Example 1 Solved

In [101]:
```
num1 = float(input('Enter a number: '))
num2 = float(input('Enter a number: '))
num3 = float(input('Enter a number: '))

#tuple has been used due to immutable nature
#the itmes are surrounded by paranthesis for addition of all numbers then division by 3
average = (num1+num2+num3)/3
print('\nThe average of the numbers is:',average)
```

```
Enter a number: 6
Enter a number: 6
Enter a number: 6

The average of the numbers is: 6.0
```

Example 2 (Logical error with if and else statement)

In [109]:

```
age = int(input("What is your age? : "))
if age <= 18: #incorrect use of logical operator
    print("You are an adult")
else:
    print("You are not an adult")
```

```
What is your age? : 23
You are not an adult
```

Example 2 solved

In [110]:

```
age = int(input("What is your age? : "))
if age >= 18: #correct use of logical operator
    print("You are an adult")
else:
    print("You are not an adult")
```

```
What is your age? : 23
You are an adult
```