Python Exception Handling

Exception can be said to be any abnormal condition in a program resulting to the disruption in the flow of the program.

Whenever an exception occurs the program halts the execution and thus further code is not executed. Thus exception is that error which python script is unable to tackle with.

Exception in a code can also be handled. In case it is not handled, then the code is not executed further and hence execution stops when exception occurs.

Common Exceptions

- **ZeroDivisionError**: Occurs when a number is divided by zero.
- **NameError**: It occurs when a name is not found. It may be local or global.
- **IndentationError**: If incorrect indentation is given.
- **IOError**: It occurs when Input Output operation fails.
- **EOFError**: It occurs when end of the file is reached and yet operations are being performed.

How to handle exception

The suspicious code can be handled by using the try block. Enclose the code which raises an exception inside the try block. The try block is followed by except statement. It is then further followed by statements which are executed during exception and in case if exception does not occur.

Syntax:

```
try:
        malicious code
except Exception1:
        execute code
except Exception2:
        execute code
....
....
except ExceptionN:
        execute code
else:
        In case of no exception, execute the else block code.
```

Example

An example where we will divide a number by Zero.

```
try:
    a = 10/0
    print(a)
except:
    print("An exception has occurred.")
else:
    print("Everything working Fine.")


# -----OUTPUT-----
An exception has occurred.
```

Explanation

The malicious code (code having exception) is enclosed in the try block.

Try block is followed by except statement. There can be multiple except statement with a single try block.

Except statement specifies the exception which occurred. In case that exception is occurred, the corresponding statement will be executed.

At the last you can provide else statement. It is executed when no exception is occurred. Python Exception(Except with no Exception)

Declaring Multiple Exception in Python

Python allows us to declare multiple exceptions using the same except statement.

```
try:
    a = 10/0
    print(a)
except ArithmeticError:
    print("An Arithmetic exception has occurred.")
except SyntaxError:
    print("An Syntax exception has occurred.")
else:
    print("Everything working Fine.")


# -----OUTPUT-----
An exception has occurred.
```

Finally Block

In case if there is any code which the user want to be executed, whether exception occurs or not then that code can be placed inside the finally block. Finally block will always be executed irrespective of the exception.

```
try:
    a = 10/0
    print(a)
except ArithmeticError:
    print("An Arithmetic exception has occurred.")
except SyntaxError:
    print("An Syntax exception has occurred.")
finally:
    print("Finally statement executed.")

# -----OUTPUT-----
An Arithmetic exception has occurred.
Finally statement executed.
```

Check which exception has occurred

Python helps out to print the Exception directly.

```
try:
    a = 10/0
    print(a)

except Exception as e:
    print(e)

# -----OUTPUT-----
division by zero
```

Raise an Exception

You can explicitly throw an exception in Python using **raise** statement. **Raise** will cause an exception to occur and thus execution control will stop in case it is not handled.

```
try:
    a = 10
    print(a)
    raise NameError("Hello")

except NameError as e:
    print("An exception occurred")
    print(e)


# -----OUTPUT-----
10
An exception occurred
Hello
```

Explanation:

1. To raise an exception, raise statement is used. It is followed by exception class name.
2. Exception can be provided with a value that can be given in the parenthesis. (Hello)
3. To access the value "as" keyword is used. "e" is used as a reference variable which stores the value of the exception.

Questions

1. Write a Python program to input a number, if it is not a number generate an error message.
2. Store the 10 questions and answer in a dictionary. Ask the questions from User, if user answers 3 wrong questions, raise an Error.