Exception Handling in Python

Student's Name

Department, Institutional Affiliation

Course Number and Name

Professor's Name

Due Date

Exception Handling in Python

An exception is an error that occurs while executing code and results in an unexpected outcome. Exception handling is a method of programmatically responding to these unexpected outcomes, which require special processing (Pierre, 2020). Exception handling in Python uses a *try-except* block. Exceptions can be built into a language or custom. The main aim of exception handling is to prevent potential failures and uncontrolled code crashes.

In Python, all built-in exceptions are derived from the *BaseException* class. The *Exception* class, which is the focus of this paper, directly derives from *BaseException.* Common exceptions encountered in day-to-day operations with python include *ValueError, IndexError, TypeError, and NameError.* Figure 1 shows which conditions may trigger these errors. To handle such exceptions, Python provides useful constructs, as shown in Figure 2. The keywords are explained as follows (Van de Klundert, n.d.):

- *try block* – contains the code monitored for the exceptions.
- *except block* – contains the code to be executed if a specific exception occurs.
- *else block* – it is executed only if no exception occurred in the *try* block.
- *finally* – it is used for clean-up. This block is always executed.

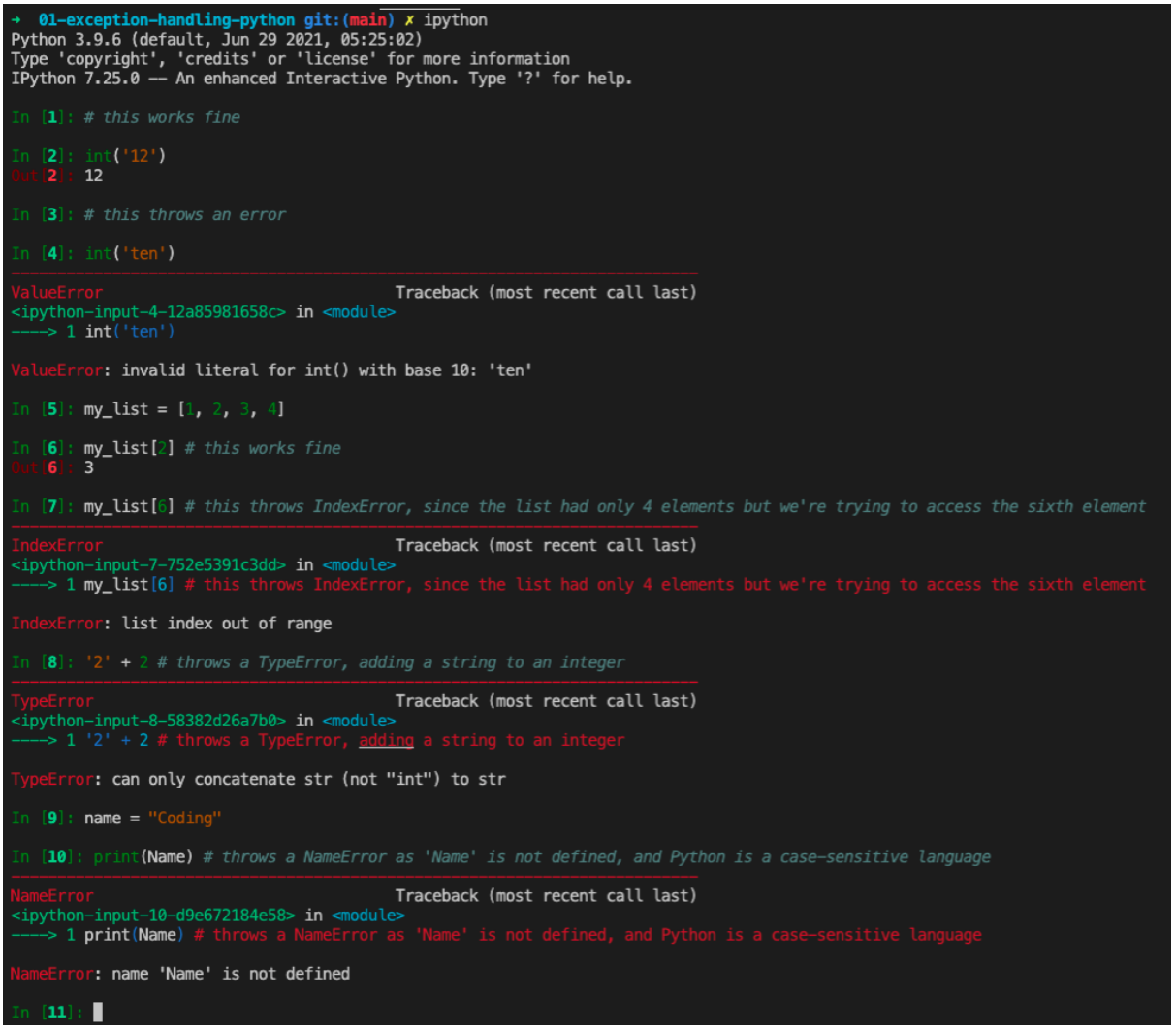
Figure 3 shows how to implement the exception handling constructs in Python. A user is required to input two numbers from the output whose sum and division are then calculated, respectively. If a string, in this case, 3, is inputted, a *TypeError* is raised. If a zero is inputted, then a *ZeroDivisionError* occurs.

An exception is an error that occurs at runtime while a code is running. Exception handling responds to these errors programmatically and makes the code continue running

without breaking. Hence, every program should have an exception handling mechanism to ensure robust and non-breaking code.

Figure 1

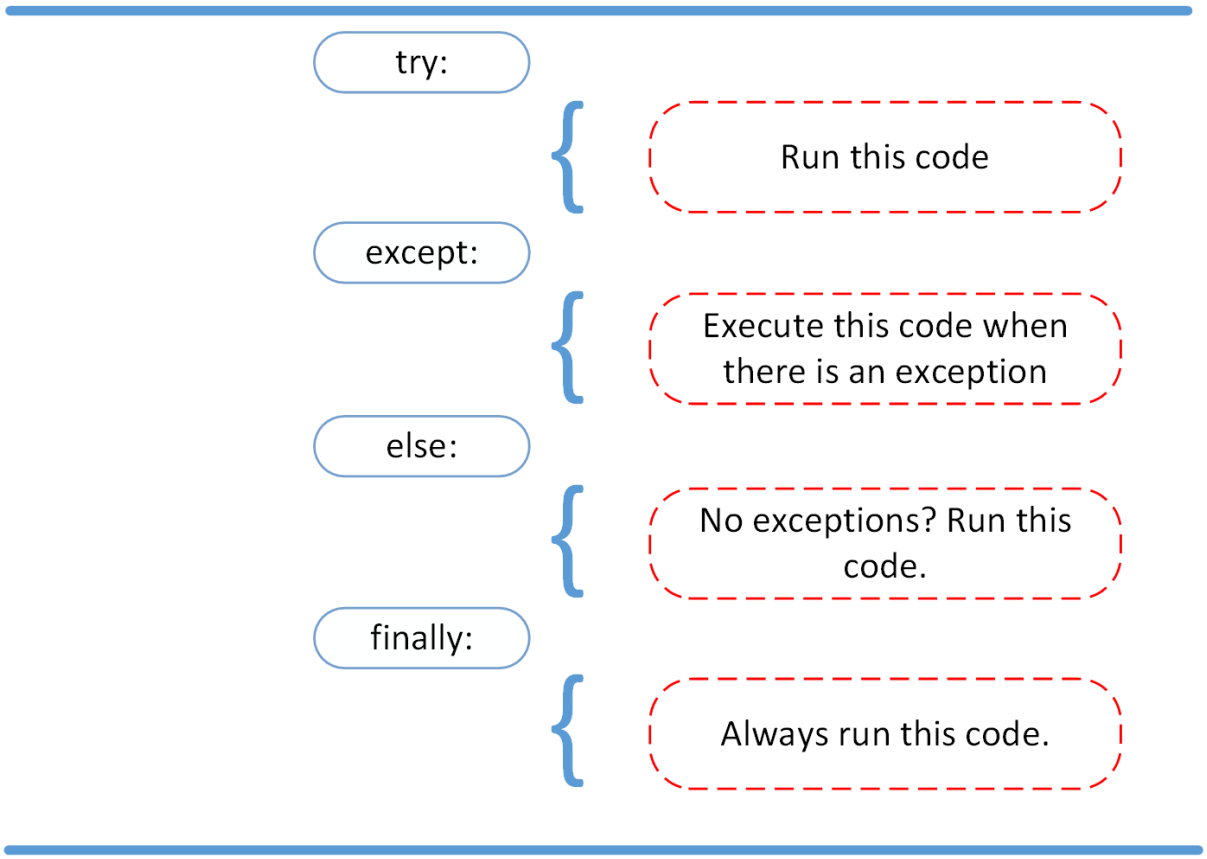
Common Python Exception



Note. Shows ValueError, IndexError, TypeError, and NameError exceptions.

Figure 2

Python Exception Handling Constructs



Note. *try*, *except*, *else* and finally *constructs* for exception handling (Van de Klundert, n.d.)

Figure 3

Python Sample Code with Common Exceptions

Note. Python code for ZeroDivisionError, ValueError, and TypeError exceptions together with their outputs.

References

Pierre, S. (2020, May 15). *Exception handling in Python.* Towards data science. https://towardsdatascience.com/exception-handling-in-python-7f639ce9a3d

Van de Klundert, S. (n.d.). *Python exceptions: An introduction*. Real Python. https://realpython.com/python-exceptions/