**John Winslow**

**11/30/2022**

**IT FDN 110**

**Assignment 07**

[**https://github.com/jwins66/IntroToProg-Python-Mod07**](https://github.com/jwins66/IntroToProg-Python-Mod07)

**Pickling and Structured Error Handling**

**Introduction:**

Pickling is a way to convert a python object (list, dict, etc.) into a character stream. This process is known as serialization. Whereby, an object in memory is converted to a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. The result can then be stored on disk or sent over a network.

Assignment 07 is an example of Pickling:

Graphical user interface, text, application, email

Description automatically generated

With the results from the Python Console:

Graphical user interface, text

Description automatically generated with medium confidence

**Questions to keep in mind and answer during the assignment:**

**What are the benefits of putting built-in Python command into functions?**

The advantages of using built-in Python functions are:

Reducing duplication of code.

Decomposing complex problems into simpler pieces.

Improving clarity of the code.

Reuse of code.

Information hiding.

<https://zetcode.com/python/function/>

**What are the benefits of using structured error handling?**

Exception handling provided a more robust mechanism for dealing with errors. Rather than instantly crashing, the error would result in an exception being raised (or thrown), allowing the programmer to catch the exception and choose how to respond to an error situation.

A benefit of exceptions in any language is that they solve the age-old problem of combining return values and error codes. Before exceptions, if an error occurred that didn’t crash the script, then the function needed a way to indicate an error.

Exceptions can return a valid value when nothing goes wrong. If something does go wrong, an exception will be raised and handled separately from the return value.

[http://conquerprogramming.com](http://conquerprogramming.com/)

**What are the differences between a text file and a binary file?**

All files can be categorized into one of two file formats – binary or text. The two file types may look the same on the surface, but they encode data differently. While both binary and text files contain data stored as a series of bits (binary values of 1s and 0s), the bits in text files represent characters, while the bits in binary files represent custom data.

While text files contain only textual data, binary files may contain both textual and custom binary data.

Below is an example of a [.PNG](https://fileinfo.com/extension/png) image file opened in an image viewer and a text editor.

|  |  |
| --- | --- |
| **Image Viewer** | **Text Editor** |
| Flower - Image Viewer | Flower - Text Editor |

As you can see, the image viewer recognizes the binary data and displays the picture. When the image is opened in a text editor, the binary data is converted to unrecognizable text. However, you may notice that some of the text is readable. This is because the PNG format includes small sections for storing textual data. The text editor, while not designed to read this file format, still displays this text when the file is opened. Many other binary file types include sections of readable text as well. Therefore, it may be possible to find out some information about an unknown binary file type by opening it in a text editor.

<https://fileinfo.com/help/binary_vs_text_files>

**How is the Exception class used?**

An exception is an error which happens at the time of execution of a program. However, while running a program, Python generates an exception that should be handled to avoid your program to crash. In Python language, exceptions trigger automatically on errors, or they can be triggered and intercepted by your code.

The exception indicates that, although the event can occur, this type of event happens infrequently. When the method is not able to handle the exception, it is thrown to its caller function. Eventually, when an exception is thrown out of the main function, the program is terminated abruptly.

<https://www.guru99.com/python-exception-handling.html>

**How do you “derive” a new class from the Exception class?**

In Python, all exceptions must be instances of a class that derives from [Base Exception](https://docs.python.org/3/library/exceptions.html#BaseException). In a [try](https://docs.python.org/3/reference/compound_stmts.html#try) statement with an [except](https://docs.python.org/3/reference/compound_stmts.html#except) clause that mentions a particular class, that clause also handles any exception classes derived from that class (but not exception classes from which it is derived). Two exception classes that are not related via subclassing are never equivalent, even if they have the same name.

The built-in exceptions listed below can be generated by the interpreter or built-in functions. Except where mentioned, they have an “associated value” indicating the detailed cause of the error. This may be a string or a tuple of several items of information (e.g., an error code and a string explaining the code). The associated value is usually passed as arguments to the exception class’s constructor.

User code can raise built-in exceptions. This can be used to test an exception handler or to report an error condition “just like” the situation in which the interpreter raises the same exception; but beware that there is nothing to prevent user code from raising an inappropriate error.

The built-in exception classes can be subclassed to define new exceptions; programmers are encouraged to derive new exceptions from the [Exception](https://docs.python.org/3/library/exceptions.html#Exception) class or one of its subclasses, and not from [Base Exception](https://docs.python.org/3/library/exceptions.html#BaseException).

<https://docs.python.org/3/library/exceptions.html>

**When might you create a class derived from the Exception class?**

To improve readability of your code.

To enhance reusability of features.

To provide custom messages/instructions to users for specific use cases.

Sometimes you are working on specific projects that require you to provide a better context into your project’s functionality. In such cases, it is better to define a custom Exception class that provides a better understanding of the errors that users can understand and relate. Using built-in exception classes may not be very useful in such scenarios.

<https://www.pylenin.com/blogs/custom-exception-class-python/>

**What is the Markdown language?**

Markdown is an easy-to-read, easy-to-write language for formatting plain text. You can use Markdown syntax, along with some additional HTML tags, to format your writing on GitHub, in places like repository READMEs and comments on pull requests and issues. In this guide, you'll learn some advanced formatting features by creating or editing a README for your GitHub profile.

<https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/quickstart-for-writing-on-github>

**How do you use Markdown on a GitHub webpage?**

# Communicate using Markdown

Organize ideas and collaborate using Markdown, a lightweight language for text formatting.

GitHub is about more than code. It’s a platform for software collaboration, and Markdown is one of the most important ways developers can make their communication clear and organized in issues and pull requests. This course will walk you through creating and using headings more effectively, organizing thoughts in bulleted lists, and showing how much work you’ve completed with checklists. You can even use Markdown to add some depth to your work with the help of emoji, images, and links.

* **Who is this for**: New developers, new GitHub users, and students.
* **What you'll learn**: Use Markdown to add lists, images, and links in a comment or text file.
* **What you'll build**: We'll update a plain text file and add Markdown formatting, and you can use this file to start your own GitHub Pages site.
* **Prerequisites**: In this course you will work with pull requests as well as edit files. If these things aren't familiar to you, we recommend you take the [Introduction to GitHub](https://github.com/skills/introduction-to-github) course, first!
* **How long**: This course is five steps long and takes less than one hour to complete.

<https://github.com/skills/communicate-using-markdown>