Programming Leftovers

By *Roy Schestowitz*
Created *17/09/2019 - 2:18am*
Submitted by Roy Schestowitz on Tuesday 17th of September 2019 02:18:06 AM Filed under Development [1]

- **post modern C tooling - draft [2]**

  Some of the C++ people have pulled off one of the cleverest and sneakiest tricks ever. They required 'modern' C99 and C11 features in 'recent' C++ standards. Microsoft has famously still clung onto some 80s version of C with their compiler for the longest time. So it's been a decade of hacks for people writing portable code in C. For a while I thought we'd be stuck in the 80s with C89 forever. However, now that some C99 and C11 features are more widely available in the Microsoft compiler, we can use these features in highly portable code (but forget about C17/C18 ISO/IEC 9899:2018/C2X stuff!!).

- **Reading and Writing YAML to a File in Python [3]**

  In this tutorial, we're going to learn how to use the YAML library in Python 3. YAML stands for Yet Another Markup Language.

  In recent years it has become very popular for its use in storing data in a serialized manner for configuration files. Since YAML essentially is a data format, the YAML library is quite brief, as the only functionality required of it is the ability to parse YAML formatted files.

  In this article we will start with seeing how data is stored in a YAML file, followed by loading that data into a Python object. Lastly, we will learn how to store a Python object in a YAML file. So, let's begin.

  Before we move further, there are a few prerequisites for this tutorial. You should have a basic understanding of Python's syntax, and/or have done at least beginner level programming experience with some other language. Other than that, the tutorial is quite simple and easy to follow for beginners.

- **Python Multiple Inheritance (with Examples)** [4]

  In this tutorial, we?ll describe Python Multiple Inheritance concept and explain how to use it in your programs. We?ll also cover multilevel inheritance, the super() function, and focus on the method resolution order.

  In the previous tutorial, we have gone through Python Class and Python (Single) Inheritance. There, you have seen that a child class inherits from a base class. However, Multiple Inheritance is a feature where a class can derive attributes and methods from more than one base classes. Hence, it creates a high level of complexity and ambiguity and known as the diamond problem in the technical world. We?ll be taking up this problem later in this tutorial.

- **Adding Methods Retroactively** [5]

  Imagine you have a "shapes" library. We have a Circle class, a Square class, etc.

  A Circle has a radius, a Square has a side, and maybe Rectangle has height and width. The library already exists: we do not want to change it.

  However, we do want to add an area calculation. If this was our library, we would just add an area method, so that we can call shape.area(), and not worry about what the shape is.

Development

Source URL: http://www.tuxmachines.org/node/128204

Links:
[1] http://www.tuxmachines.org/taxonomy/term/145
[2] http://renesd.blogspot.com/2019/09/post-modern-c-tooling.html
[3] https://stackabuse.com/reading-and-writing-yaml-to-a-file-in-python/
[4] https://www.techbeamers.com/python-multiple-inheritance/
[5] https://orbifold.xyz/singledispatch.html