CSCI 39538: Pep Prep Python Out of 15 points

**Background:**

[Python PEPs](https://peps.python.org/) are proposals made by the Python Dev team to propose enhancements to the language. They’ve been around since 2000 and the stand for Python Enhancement Proposals. PEPs are useful to study because they’re great examples of design docs made by skilled developers (you’ve got to be pretty good if you’re on Python’s dev council). They’re generally written in a concise fashion with specific technical feature specifications and they’re the mechanism for deciding and designing new features in future Python releases.

Besides documenting potential new features, it’s a place for community members (generally open-source contributors of larger Python projects) to give input on an issue, a place to note various limitations and a central place where community consensus can be gathered.

I note here that there are multiple ‘versions’ of Python, meaning that besides Python 2 and Python 3, there are versions of Python which are not based on C. For the purposes of this project you’ll be working with CPython, which is the version of Python that the majority of builds are based on. The core developers of CPython, the CPython Steering Council and developers of other Python implementations are the target audience of PEPs. There is also room for other members of the Python community to document API conventions and manage complex design problems.

There are [three kinds of PEPs](https://peps.python.org/pep-0001/#pep-types), the Standards Track, Informational and Process PEPs.

A Standards Track PEP describes new features or implementations.

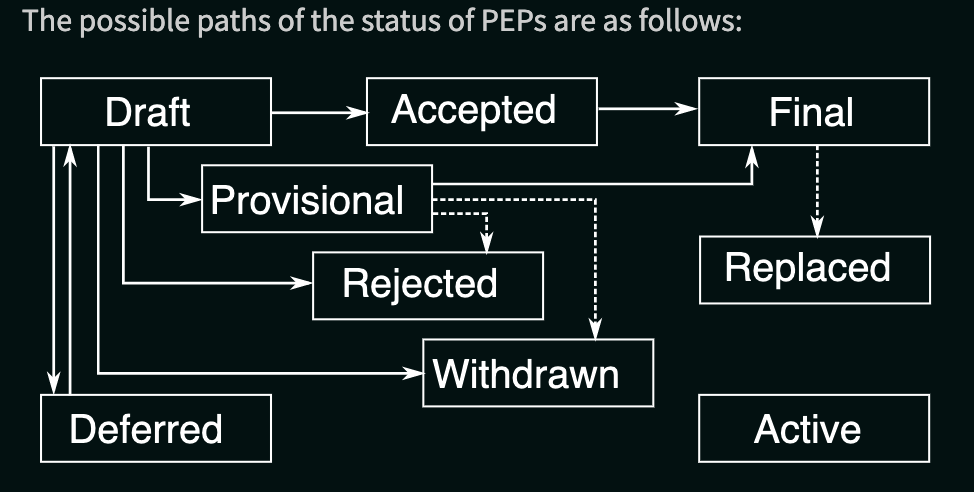
An Informational PEP describes a Python design issue or general guidelines/information. T does not propose a new feature. These tend to be ‘recommendations’ for Python developers

A Process PEP describes a process surrounding Python but does not suggest a change to the Python language itself. They are often more than recommendations and they involve changes to tools or environments used in Python development.

In the past, the final decision was left to Python’s Benevolent Dictator For Life, Guido Van Rossum (who invented Python over Winter Break in the 90s). Van Rossum stepped down from the post after a controversial decision he made a few years ago (Instead of a BNFL, a PEP‑Delegate is nominated who has similar permissions). Now the Steering Council and the Core Developers are the members who control whether a PEP is accepted or how it may be modified (you can check a PEP’s history in a way similar to GitHub’s commit system). There are PEP editors who manage the PEP throughout the PEP workflow.

A PEP begins as a new idea for Python and a single PEP generally contains a single idea. If a PEP is unfocused it can often be rejected. Each PEP has a champion or someone who makes sure it follows the PEP style and guidelines (in other words, the author). It pays to ask the Python community about an issue before submitting a PEP, it will reduce the chance of the PEP being rejected.

Changing Python is complicated, it’s not as bad as submitting a bill to congress and passing a law but it is hard.



**Project Description:**

There are three kinds of PEPs in Python, your job is to select four accepted PEPs, two Standards Tracks, 1 Information and one Process and write up why these PEPs were necessary and accepted, the discussion/methodology behind the PEP (look at the motivation/rationale sections) and what benefits/negatives the PEP brought to Python.

Then you need to write your own PEP for a feature/information/process you’d like to add/contribute/change in Python. This PEP needs to follow the conventions [here](https://peps.python.org/pep-0001/#what-belongs-in-a-successful-pep).

There’s a lot to gain from learning how to read and write technical documents and it’s a skill not covered enough in university. When you finish the project let me know if you found this project useful (you can write it on your response in BB).

**Rubric:**

* (10 points) Each of the four PEPs you need to research will be worth 2.5 points
  + .5 points for why the PEP was necessary
  + 2 points for your discussion of the PEP using any of the resources in the PEP (there are 13 parts of a PEP so feel free to use whichever parts of it you want)
* (5 points) The section where you write your own PEP will be worth 5 points and it needs a preamble, abstract, motivation and specification
* (EC 3 points) You can get up to 3 bonus points on this project by finding up to 3 rejected PEPs and writing about why they were rejected