Term Project

Author: Khasanboy Khakimjanov

**Project Description**

The goal of this project is to develop a Python code style checker based on the 15-112 style guides. This tool will:

* Provide users with detailed explanations of styling violations, pinpointing the exact line of code where they occur.
* Offer users the option to select specific styling errors to check for.
* Include an interactive styling guide that navigates users directly to the code line associated with a detected styling issue.

**Competitive Analysis**

* While this project shares similarities with existing tools like Gradescope, it aims to surpass them by incorporating a broader range of features.

**Structure Plan**

* The design will feature separate classes for each widget displayed on the screen, enhancing reusability.
* The project will be modular, with divisions based on the types of components, widgets, and logical operations.
* A dedicated class will be implemented for style checking, facilitating better project structure management and accessibility.
* All components and logical operations will be integrated into the main Python file.

**Algorithmic Plan**

* The main challenge involves extracting identifiers from the provided Python code. For this, I'll utilize the AST (Abstract Syntax Trees) - a built-in Python library that aids in extracting functions, classes, variables, etc.

**Version Control**

* Regular updates and commits will be made to GitHub for backup purposes. The project can be accessed at: GitHub Repository

**Timeline**

* 14 November: Development of a basic checker to display styling errors by entering the Python code path in the shell.
* 21 November: Enhancement with more interactive code and error displays, including scrollbars and an internal file explorer for opening Python code.
* 30 November: Introduction of features like selectable styling criteria, test case creation for functions, and expanded styling options.

**Possible Features**

* Implementation of colored code snippets for differentiating Python code.
* Test case checker.

**Module List**

* Builtin AST (Abstract Syntax Tree) (built-in)
* CMU Graphics