

IT314 - Software Engineering
Lab 5 - Static Analysis

Dev Jadav
202001016

Tools:

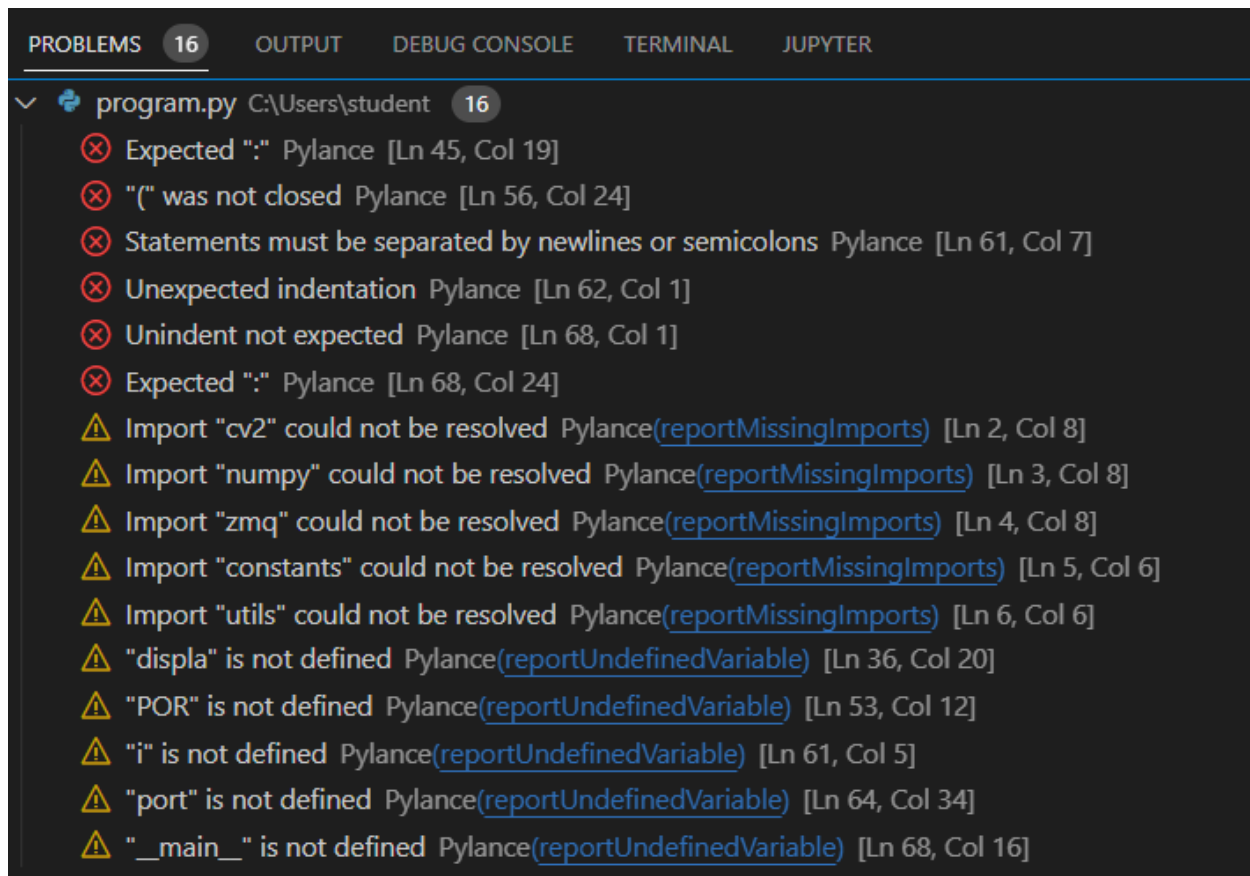
- Mypy: <https://github.com/python/mypy>
Mypy is a static type checker for Python. It analyzes your Python code and determines the type of each variable, function parameter, and return value, and then checks that they are being used correctly throughout the code. This helps catch errors early in the development process, and also makes code easier to understand and maintain.
- Pylint: <https://github.com/PyCQA/pylint>
Pylint is a tool for checking your Python code for errors, code style violations, and other issues. It checks for issues such as unused variables, undefined variables, invalid function names, and more. It also provides suggestions for improving the code, such as suggesting more descriptive variable names or recommending the use of list comprehensions.
- Pyflakes: <https://github.com/PyCQA/pyflakes>
Pyflakes is another Python code checker that focuses on finding common errors and issues in your code. It checks for issues such as undefined variables, unused imports, and other common mistakes. Pyflakes is lightweight and fast, making it a good choice for quick code analysis.
- Pycodestyle (pep8): <https://github.com/PyCQA/pycodestyle>
Pycodestyle (formerly known as pep8) is a tool for checking your Python code against the PEP 8 style guide. PEP 8 is a set of guidelines for writing Python code that is easy to read and maintain. Pycodestyle checks your code for violations of these guidelines, such as using tabs instead of spaces, having lines that are too long, and more.
- Flake8: <https://github.com/PyCQA/flake8>
Flake8 is a combination of Pyflakes, Pycodestyle, and McCabe complexity checker. It checks for errors, code style violations, and complexity issues all in one tool. It's a popular choice for Python code checking because it combines multiple tools into one easy-to-use package.
- Prospector: <https://github.com/PyCQA/prospector>
Prospector is a tool that combines several other Python code analysis tools, including Pylint, Pyflakes, and pep8. It also includes additional checks for issues such as security vulnerabilities and code smells. Prospector provides a comprehensive analysis of your Python code and can help you identify areas for improvement.
- Bandit: <https://github.com/PyCQA/bandit>
Bandit is a security-focused tool for checking your Python code for potential security vulnerabilities. It analyzes your code for issues such as hardcoded passwords, SQL injection vulnerabilities, and more. Bandit is an important tool for ensuring that your Python code is secure and free from common security risks.

Code:

The following code is taken from the Github repository SmoothStream
<https://github.com/CT83/SmoothStream/blob/master/StreamViewer.py>

Tool used: **Pylint**

Erros:



The screenshot shows a code editor interface with a dark theme. At the top, there is a tab bar with 'PROBLEMS' selected, showing a count of 16. Below the tab bar, the file 'program.py' is open, located at 'C:\Users\student'. The list of errors is as follows:

- Expected ":" Pylance [Ln 45, Col 19]
- "(" was not closed Pylance [Ln 56, Col 24]
- Statements must be separated by newlines or semicolons Pylance [Ln 61, Col 7]
- Unexpected indentation Pylance [Ln 62, Col 1]
- Unindent not expected Pylance [Ln 68, Col 1]
- Expected ":" Pylance [Ln 68, Col 24]
- Import "cv2" could not be resolved Pylance(reportMissingImports) [Ln 2, Col 8]
- Import "numpy" could not be resolved Pylance(reportMissingImports) [Ln 3, Col 8]
- Import "zmq" could not be resolved Pylance(reportMissingImports) [Ln 4, Col 8]
- Import "constants" could not be resolved Pylance(reportMissingImports) [Ln 5, Col 6]
- Import "utils" could not be resolved Pylance(reportMissingImports) [Ln 6, Col 6]
- "displa" is not defined Pylance(reportUndefinedVariable) [Ln 36, Col 20]
- "POR" is not defined Pylance(reportUndefinedVariable) [Ln 53, Col 12]
- "i" is not defined Pylance(reportUndefinedVariable) [Ln 61, Col 5]
- "port" is not defined Pylance(reportUndefinedVariable) [Ln 64, Col 34]
- "__main__" is not defined Pylance(reportUndefinedVariable) [Ln 68, Col 16]

Code Snippets:

```
import argparse
import cv2
import numpy as np
import zmq
from constants import PORT
from utils import string_to_image
```

```
45     def stop(self):
46         """
47         Sets 'keep_running' to False to stop the running loop if running.
48         :return: None
49         """
50         self.keep_running = False
51
52     def main():
53         port = PORT
54
55         parser = argparse.ArgumentParser()
56         parser.add_argument('-p', '--port'
57                             help='The port which you want the Streaming Viewer to use, default'
58                             ' is ' + PORT, required=False)
59
60         args = parser.parse_args()
61         if args.port
62             port = args.port
63
64         stream_viewer = StreamViewer(port)
65         stream_viewer.receive_stream()
66
67
68     if __name__ == '__main__':
69         main()
```