

**Name:** Bhavya Rajdev

**Id:** 202001067

## **IT314 Lab-5**

### **Static analysis:**

Static analysis is a method for examining source code without executing it. It means to check for errors, bugs, vulnerabilities, and other quality issues in the code.

Tools for static analysis can perform various tasks such as checking syntax, style, logic, data flow, control flow, and security.

They can also measure various metrics for the code, such as complexity, readability, maintainability, test coverage, and documentation.

This can improve the reliability, performance, and maintainability of software by identifying and correcting defects early in the development process.

### **Pylint:**

Pylint is a static code analyser for Python 2 or 3. It analyses the code without actually running it. It checks for errors, enforces a coding standard, looks for code smells, and can make suggestions about how the code could be refactored. Pylint can also be integrated with various editors or IDEs

### **Running pylint on different python files to see the output:**

1> This file calculates the age for a given birth date on GUI

**URL of file:** [https://github.com/avinashkranjan/Amazing-Python-Scripts/blob/master/Age-Calculator-GUI/age\\_calc\\_gui.py](https://github.com/avinashkranjan/Amazing-Python-Scripts/blob/master/Age-Calculator-GUI/age_calc_gui.py)

## Output of pylint:

```
check.py:56:0: C0301: Line too long (115/100) (line-too-long)
check.py:63:0: C0301: Line too long (119/100) (line-too-long)
check.py:74:0: C0301: Line too long (120/100) (line-too-long)
check.py:81:0: C0301: Line too long (120/100) (line-too-long)
check.py:92:0: C0301: Line too long (118/100) (line-too-long)
check.py:99:0: C0301: Line too long (120/100) (line-too-long)
check.py:113:0: C0301: Line too long (113/100) (line-too-long)
check.py:115:0: C0301: Line too long (117/100) (line-too-long)
check.py:119:0: C0301: Line too long (104/100) (line-too-long)
check.py:1:0: C0114: Missing module docstring (missing-module-docstring)
check.py:8:0: C0115: Missing class docstring (missing-class-docstring)
check.py:20:8: C0103: Attribute name "l1" doesn't conform to snake_case naming style (invalid-name)
check.py:25:8: C0103: Attribute name "nameEntry" doesn't conform to snake_case naming style (invalid-name)
check.py:29:8: C0103: Attribute name "l2" doesn't conform to snake_case naming style (invalid-name)
check.py:32:8: C0103: Attribute name "yearEntry" doesn't conform to snake_case naming style (invalid-name)
check.py:36:8: C0103: Attribute name "l3" doesn't conform to snake_case naming style (invalid-name)
check.py:39:8: C0103: Attribute name "monthEntry" doesn't conform to snake_case naming style (invalid-name)
check.py:43:8: C0103: Attribute name "l4" doesn't conform to snake_case naming style (invalid-name)
check.py:46:8: C0103: Attribute name "dayEntry" doesn't conform to snake_case naming style (invalid-name)
check.py:8:0: R0902: Too many instance attributes (11/7) (too-many-instance-attributes)
check.py:18:4: C0116: Missing function or method docstring (missing-function-docstring)
check.py:22:8: C0103: Variable name "nameValue" doesn't conform to snake_case naming style (invalid-name)
check.py:31:8: C0103: Variable name "yearValue" doesn't conform to snake_case naming style (invalid-name)
check.py:38:8: C0103: Variable name "monthValue" doesn't conform to snake_case naming style (invalid-name)
check.py:45:8: C0103: Variable name "dayValue" doesn't conform to snake_case naming style (invalid-name)
check.py:62:19: W0718: Catching too general exception Exception (broad-exception-caught)
check.py:55:16: R1705: Unnecessary "else" after "return", remove the "else" and de-indent the code inside it (no-else-return)
check.py:62:12: C0103: Variable name "e" doesn't conform to snake_case naming style (invalid-name)
check.py:62:12: W0612: Unused variable 'e' (unused-variable)
check.py:80:19: W0718: Catching too general exception Exception (broad-exception-caught)
check.py:73:16: R1705: Unnecessary "else" after "return", remove the "else" and de-indent the code inside it (no-else-return)
```

check.py:80:12: C0103: Variable name "e" doesn't conform to snake\_case naming style (invalid-name)  
check.py:80:12: W0612: Unused variable 'e' (unused-variable)  
check.py:98:19: W0718: Catching too general exception Exception (broad-exception-caught)  
check.py:91:16: R1705: Unnecessary "else" after "return", remove the "else" and de-indent the code inside it (no-else-return)  
check.py:98:12: C0103: Variable name "e" doesn't conform to snake\_case naming style (invalid-name)  
check.py:98:12: W0612: Unused variable 'e' (unused-variable)  
check.py:105:8: C0103: Function name "ageCalc" doesn't conform to snake\_case naming style (invalid-name)  
check.py:110:16: C0103: Variable name "birthDate" doesn't conform to snake\_case naming style (invalid-name)  
check.py:18:4: R0915: Too many statements (64/50) (too-many-statements)  
check.py:20:8: W0201: Attribute 'l1' defined outside \_\_init\_\_ (attribute-defined-outside-init)  
check.py:25:8: W0201: Attribute 'nameEntry' defined outside \_\_init\_\_ (attribute-defined-outside-init)  
check.py:29:8: W0201: Attribute 'l2' defined outside \_\_init\_\_ (attribute-defined-outside-init)  
check.py:32:8: W0201: Attribute 'yearEntry' defined outside \_\_init\_\_ (attribute-defined-outside-init)  
check.py:36:8: W0201: Attribute 'l3' defined outside \_\_init\_\_ (attribute-defined-outside-init)  
check.py:39:8: W0201: Attribute 'monthEntry' defined outside \_\_init\_\_ (attribute-defined-outside-init)  
check.py:43:8: W0201: Attribute 'l4' defined outside \_\_init\_\_ (attribute-defined-outside-init)  
check.py:46:8: W0201: Attribute 'dayEntry' defined outside \_\_init\_\_ (attribute-defined-outside-init)  
check.py:119:8: W0201: Attribute 'button' defined outside \_\_init\_\_ (attribute-defined-outside-init)

-----  
Your code has been rated at 4.25/10 (previous run: 4.25/10, +0.00)

As the code is correct, above output is not showing any errors but it is showing some code style improvements by showing warnings for naming conventions, adding module doc string, not writing too big lines and not to define an attribute of a class outside init method because this practice might be inconvenient and others might have problems identifying attributes of a class.

2> This file retrieves the price of the product from amazon in specified time interval

**URL of the file:** <https://github.com/avinashkranjan/Amazing-Python-Scripts/blob/master/Amazon-Price-Tracker/amazonprice.py>

### Output of pylint:

\*\*\*\*\* Module check

check.py:7:0: C0301: Line too long (119/100) (line-too-long)

check.py:10:0: C0301: Line too long (422/100) (line-too-long)

check.py:31:0: C0301: Line too long (124/100) (line-too-long)

check.py:77:0: C0325: Unnecessary parens after '=' keyword (superfluous-parens)

check.py:95:0: C0304: Final newline missing (missing-final-newline)

check.py:1:0: C0114: Missing module docstring (missing-module-docstring)

check.py:17:7: W3101: Missing timeout argument for method 'requests.get' can cause your program to hang indefinitely (missing-timeout)

check.py:23:0: C0116: Missing function or method docstring (missing-function-docstring)

check.py:37:0: C0116: Missing function or method docstring (missing-function-docstring)

check.py:46:4: W0702: No exception type(s) specified (bare-except)

check.py:55:8: W0702: No exception type(s) specified (bare-except)

check.py:65:12: W0702: No exception type(s) specified (bare-except)

check.py:87:8: R1722: Consider using 'sys.exit' instead (consider-using-sys-exit)

check.py:3:0: C0411: standard import "import time" should be placed before "import requests" (wrong-import-order)

check.py:4:0: C0411: standard import "import smtplib" should be placed before "import requests" (wrong-import-order)

-----

Your code has been rated at 6.88/10 (previous run: 4.25/10, +2.62)

This output is showing few extra messages than previous execution. Warnings are shown for unnecessary parenthesis, missing function docstring, missing final newline, missing the specific type of except, suggests to use sys.exit() instead of exit(), suggests for better import orderings, also suggests to write a timeout exception for get method of requests.

3> Base\_N calculator is a GUI that takes any number of any base as input and converts it to required base number.

**URL of the file:** [https://github.com/avinashkranjan/Amazing-Python-Scripts/blob/master/Base-N\\_Calc/script.py](https://github.com/avinashkranjan/Amazing-Python-Scripts/blob/master/Base-N_Calc/script.py)

### Output of pylint:

\*\*\*\*\* Module check

check.py:11:0: C0301: Line too long (105/100) (line-too-long)

check.py:21:0: C0301: Line too long (116/100) (line-too-long)

check.py:40:0: C0301: Line too long (107/100) (line-too-long)

check.py:45:0: C0325: Unnecessary parens after 'while' keyword (superfluous-parens)

check.py:47:0: C0301: Line too long (116/100) (line-too-long)

check.py:154:0: C0301: Line too long (118/100) (line-too-long)

check.py:159:0: C0304: Final newline missing (missing-final-newline)

check.py:5:42: W1401: Anomalous backslash in string: '\ '. String constant might be missing an r prefix. (anomalous-backslash-in-string)

check.py:1:0: C0114: Missing module docstring (missing-module-docstring)

check.py:1:0: W0401: Wildcard import tkinter (wildcard-import)

check.py:3:0: C0103: Function name "SetStatusError" doesn't conform to snake\_case naming style (invalid-name)

check.py:9:0: C0116: Missing function or method docstring (missing-function-docstring)

check.py:9:0: C0103: Function name "Con\_Base\_X\_to\_Dec" doesn't conform to snake\_case naming style (invalid-name)

check.py:1:0: W0614: Unused import(s) enum, sys, types, TclError, re, wantobjects, TkVersion, TclVersion, READABLE, WRITABLE, EXCEPTION, EventType, Event, NoDefaultRoot, Variable, IntVar, DoubleVar, BooleanVar,.mainloop, getint, getdouble, getboolean, Misc, CallWrapper, XView, YView, Wm, Tcl, Pack, Place, Grid, BaseWidget, Widget, Toplevel, Canvas, Checkbutton, Frame, Listbox, Menu, Menubutton, Message, Radiobutton, Scale, Scrollbar, Text, OptionMenu, Image, BitmapImage, image\_names, image\_types, Spinbox, LabelFrame, PanedWindow, NO, FALSE, OFF, YES, TRUE, ON, N, S, NW, SW, NE, SE, NS, EW, NSEW, CENTER, NONE, Y, BOTH, LEFT, TOP, RIGHT, RAISED, FLAT, RIDGE, GROOVE, SOLID, HORIZONTAL, VERTICAL, NUMERIC, CHAR, WORD, BASELINE, INSIDE, OUTSIDE, SEL, SEL\_FIRST, SEL\_LAST, END, INSERT, CURRENT, ANCHOR, ALL, NORMAL, DISABLED, ACTIVE, HIDDEN, CASCADE, CHECKBUTTON, COMMAND, RADIOBUTTON, SEPARATOR, SINGLE, BROWSE, MULTIPLE, EXTENDED, DOTBOX, UNDERLINE, PIESLICE, CHORD, ARC, FIRST, LAST, BUTT, PROJECTING, ROUND, BEVEL, MITER, MOVETO, SCROLL, UNITS and PAGES from wildcard import of tkinter (unused-wildcard-import)

-----  
Your code has been rated at 7.71/10 (previous run: 6.88/10, +0.84)

In this code execution, change from previous execution is that it suggests to not all the functions from some module if not required, also suggests to not keep the anomalous backslash in the string.

4> This file is bubble sort visualizer. It opens GUI and allow us to set various parameters of the array and starts visualization of sorting the array.

**URL of the file:** [https://github.com/avinashkranjan/Amazing-Python-Scripts/blob/master/Bubble-Sort-Visualization/bubble\\_sort.py](https://github.com/avinashkranjan/Amazing-Python-Scripts/blob/master/Bubble-Sort-Visualization/bubble_sort.py)

### Output of pylint:

\*\*\*\*\* Module check

check.py:46:0: C0301: Line too long (105/100) (line-too-long)

check.py:56:0: C0325: Unnecessary parens after 'if' keyword (superfluous-parens)

check.py:157:0: C0304: Final newline missing (missing-final-newline)

check.py:1:0: C0114: Missing module docstring (missing-module-docstring)

check.py:1:0: W0401: Wildcard import tkinter (wildcard-import)

check.py:9:14: W0621: Redefining name 'data' from outer scope (line 6) (redefined-outer-name)

check.py:49:4: W0603: Using the global statement (global-statement)

check.py:63:8: W0612: Unused variable 'i' (unused-variable)

check.py:75:4: W0603: Using the global statement (global-statement)

check.py:80:16: W0621: Redefining name 'data' from outer scope (line 6) (redefined-outer-name)

check.py:92:41: R1714: Consider merging these comparisons with 'in' by using 'x in (j, j + 1)'. Use a set instead if elements are hashable. (consider-using-in)

check.py:104:4: W0602: Using global for 'data' but no assignment is done (global-variable-not-assigned)

check.py:1:0: W0614: Unused import(s) enum, sys, types, TclError, re, wantobjects, TkVersion, TclVersion, READABLE, WRITABLE, EXCEPTION, EventType, Event, NoDefaultRoot, Variable, StringVar, IntVar, DoubleVar, BooleanVar, mainloop, getint, getdouble, getboolean, Misc, CallWrapper, XView, YView, Wm, Tcl, Pack, Place, Grid, BaseWidget, Widget, Toplevel, Checkbutton, Entry, Listbox, Menu, Menubutton, Message, Radiobutton, Scrollbar, Text, OptionMenu, Image, PhotoImage, BitmapImage, image\_na

mes, image\_types, Spinbox, LabelFrame, PanedWindow, NO, FALSE, OFF, YES, TRUE, ON, N, S, E, NW, NE, SE, NS, EW, NSEW, CENTER, NONE, X, Y, BOTH, LEFT, TOP, RIGHT, BO

TTOM, RAISED, SUNKEN, FLAT, RIDGE, GROOVE, SOLID, VERTICAL, NUMERIC, CHAR, WORD, BASELINE, INSIDE, OUTSIDE, SEL, SEL\_FIRST, SEL\_LAST, END, INSERT, CURRENT, ANCHOR,

SEPARATOR, SINGLE, BROWSE, MULTIPLE, EXTENDED, DOTBOX, UNDERLINE, PIESLICE, CHORD, ARC, FIRST, LAST, BUTT, PROJECTING, ROUND, BEVEL, MITER, MOVETO, SCROLL, UNITS and PAGES from wildcard import of tkinter (unused-wildcard-import)

-----  
Your code has been rated at 8.14/10 (previous run: 8.14/10, +0.00)

More from the previous executions pylint also suggests warnings for unused variables and warnings for using global statements.

As we can see that for different files, it is giving a lot of suggestions, warnings and suggestions to write better code. It could be very useful for checking the bugs in the code before even running the code and also help us improve write better styled code.