# Python Speedtest Tutorial

## Contents

Python Speedtest Tutorial ........ 1

Speedtest-cli ........ 1

Tutorial 1: Research ........ 1

Tutorial 2: Speedtest CLI Beta ........ 2

Tutorial 3: Speedtest CLI Simple ........ 3

Tutorial 4: Speedtest CLI OOP ........ 5

Tutorial 5: Speedtest GUI Tkinter ........ 8

Assignment Submission ........ 12

Time required: 90 minutes

# Speedtest-cli

Speedtest.net that tests the speed and performance of your internet connection. Speedtest is available in many forms, web site, apps, and a Python library located at https://pypi.org/project/speedtest-cli

The **speedtest-cli** Python library provides a command line interface (CLI) for testing internet bandwidth using speedtest.net.

The following tutorials show a development process from research to testing to final product.

# Tutorial 1: Research

We want to be able to measure internet bandwidth from a Python program.

1. Google **python speedtest**
2. The first result usually is a CLI application from Speedtest.net. This is the type of program we are looking for. https://www.speedtest.net/apps/cli
3. The next result is usually a Python library, https://pypi.org/project/speedtest-cli/ A step in the right direction. We have found a Python library that wraps speedtest-cli program.

4. Let’s look for some tutorials to get a better idea of how this library works. We will search for Python speedtest-cli tutorials or python internet speedtest.

    a. https://www.geeksforgeeks.org/test-internet-speed-using-python/

    b. https://yourblogcoach.com/how-to-test-internet-speed-using-python/

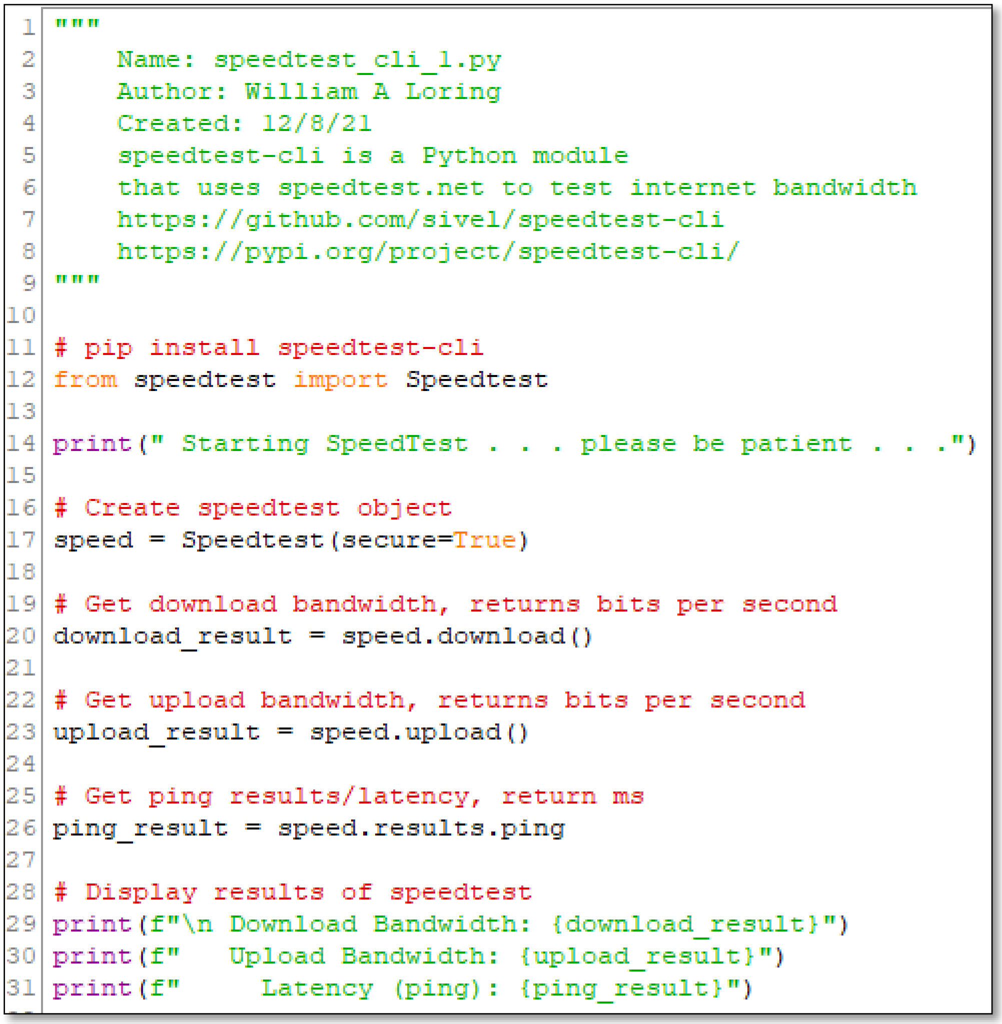
    c. https://pyshark.com/test-internet-speed-using-python/

    d. https://www.codespeedy.com/test-internet-speed-using-python/

## Tutorial 2: Speedtest CLI Beta

This is the first version after researching tutorials and speed-cli documentation. This version is to make sure the basic methods are working.

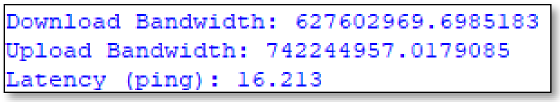
Create a Python program named **speedtest_cli_1.py**



```
"""
    Name: speedtest_cli_1.py
    Author: William A Loring
    Created: 12/8/21
    speedtest-cli is a Python module
    that uses speedtest.net to test internet bandwidth
    https://github.com/sivel/speedtest-cli
    https://pypi.org/project/speedtest-cli/
"""

# pip install speedtest-cli
from speedtest import Speedtest

print(" Starting SpeedTest . . . please be patient . . .")

# Create speedtest object
speed = Speedtest(secure=True)

# Get download bandwidth, returns bits per second
download_result = speed.download()

# Get upload bandwidth, returns bits per second
upload_result = speed.upload()

# Get ping results/latency, return ms
ping_result = speed.results.ping

# Display results of speedtest
print(f"\n Download Bandwidth: {download_result}")
print(f"   Upload Bandwidth: {upload_result}")
print(f"    Latency (ping): {ping_result}")
```

There isn’t any feedback on this program until the end. Be patient, it can take almost 30 seconds to finish the test.

Example run:



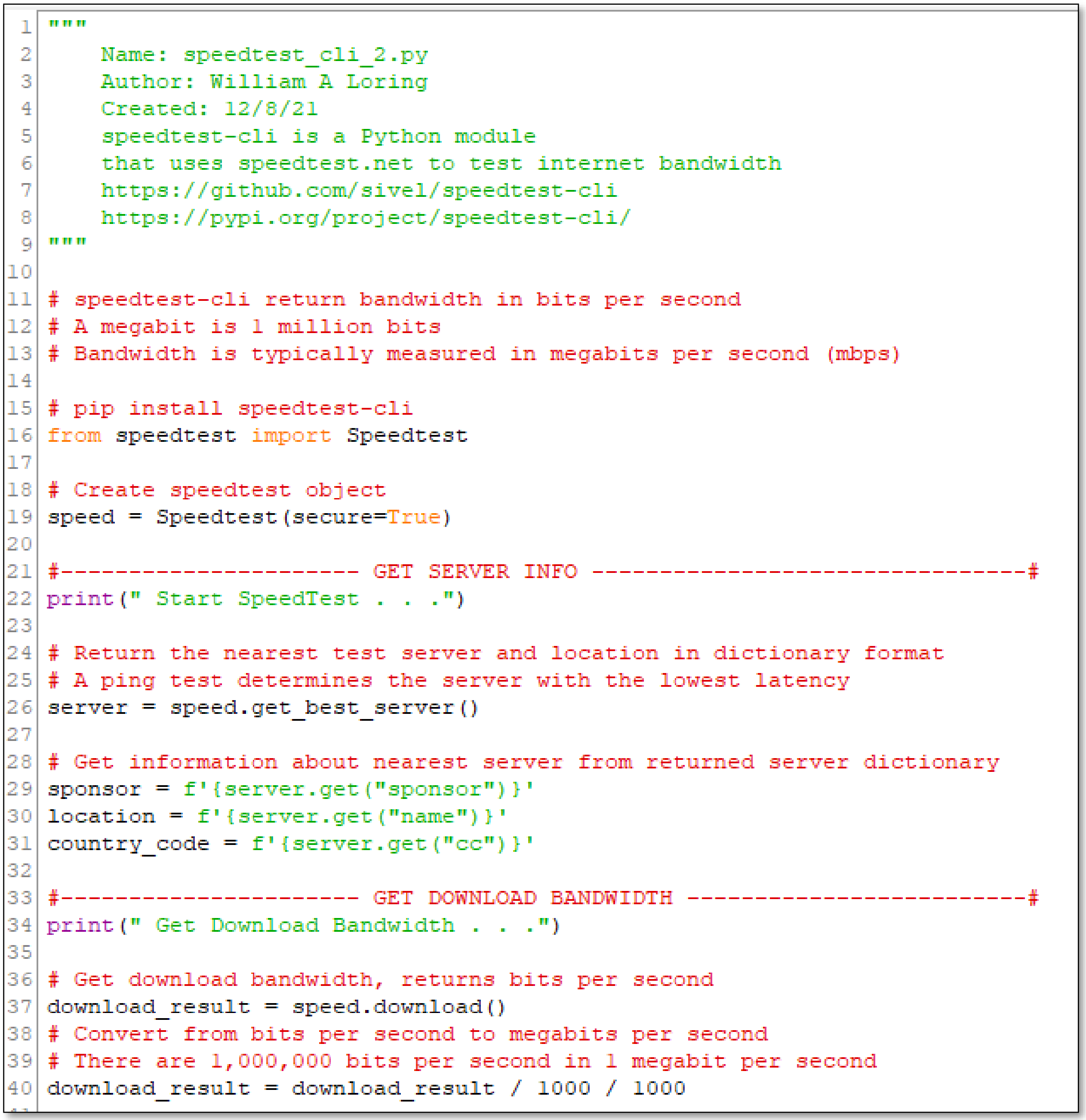
```
Download Bandwidth: 627602969.6985183
Upload Bandwidth: 742244957.0179085
Latency (ping): 16.213
```

## Tutorial 3: Speedtest CLI Simple

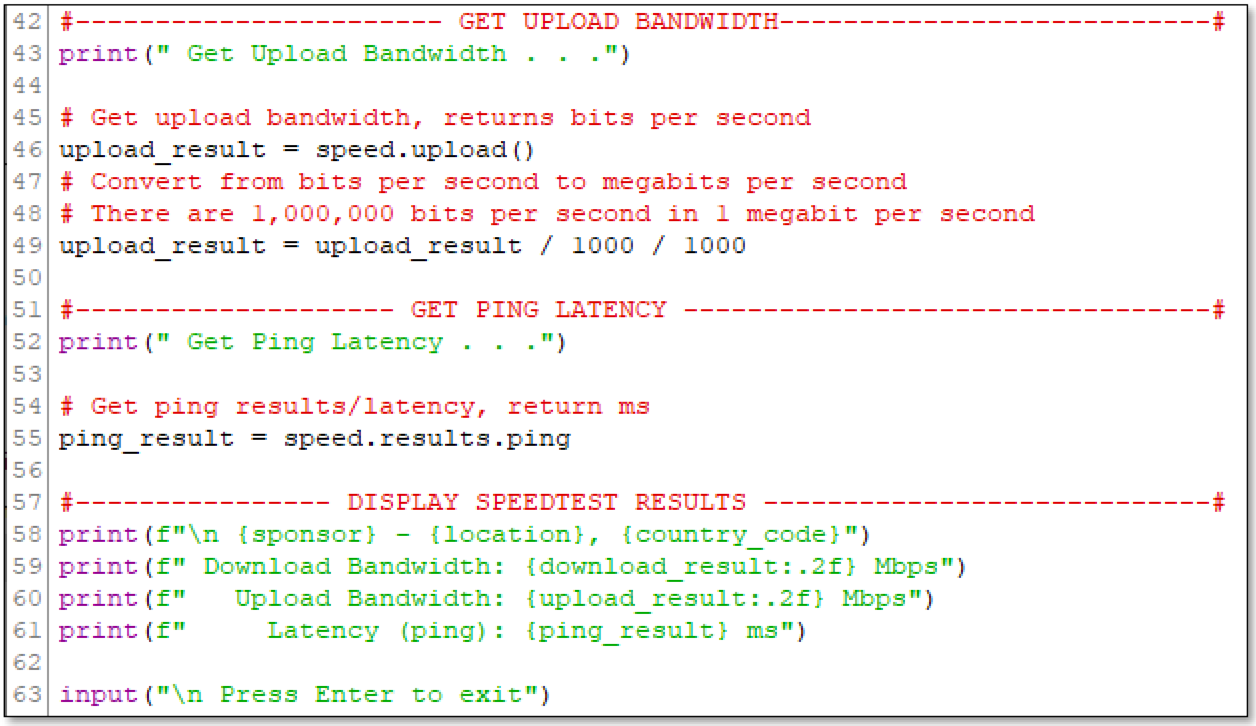
Copy the last file and rename it as **speedtest_cli_2.py**

Keeping previous versions of programs is helpful if the changes you are making break the program.

This version added conversion to megabits per second and some display additions.

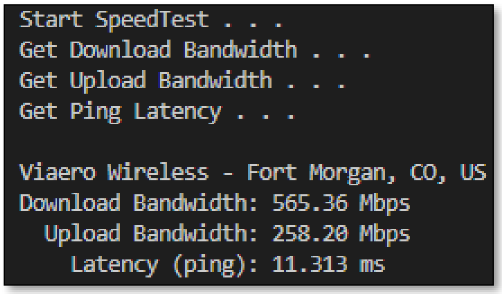


```
"""
    Name: speedtest_cli_2.py
    Author: William A Loring
    Created: 12/8/21
    speedtest-cli is a Python module
    that uses speedtest.net to test internet bandwidth
    https://github.com/sivel/speedtest-cli
    https://pypi.org/project/speedtest-cli/
"""

# speedtest-cli return bandwidth in bits per second
# A megabit is 1 million bits
# Bandwidth is typically measured in megabits per second (mbps)

# pip install speedtest-cli
from speedtest import Speedtest

# Create speedtest object
speed = Speedtest(secure=True)

#---------------------- GET SERVER INFO --------------------------------#
print(" Start SpeedTest . . .")

# Return the nearest test server and location in dictionary format
# A ping test determines the server with the lowest latency
server = speed.get_best_server()

# Get information about nearest server from returned server dictionary
sponsor = f'{server.get("sponsor")}'
location = f'{server.get("name")}'
country_code = f'{server.get("cc")}'

#---------------------- GET DOWNLOAD BANDWIDTH -------------------------#
print(" Get Download Bandwidth . . .")

# Get download bandwidth, returns bits per second
download_result = speed.download()
# Convert from bits per second to megabits per second
# There are 1,000,000 bits per second in 1 megabit per second
download_result = download_result / 1000 / 1000
```



```python
#------------------------ GET UPLOAD BANDWIDTH---------------------------#
print(" Get Upload Bandwidth . . .")

# Get upload bandwidth, returns bits per second
upload_result = speed.upload()
# Convert from bits per second to megabits per second
# There are 1,000,000 bits per second in 1 megabit per second
upload_result = upload_result / 1000 / 1000

#-------------------- GET PING LATENCY ---------------------------------#
print(" Get Ping Latency . . .")

# Get ping results/latency, return ms
ping_result = speed.results.ping

#---------------- DISPLAY SPEEDTEST RESULTS ----------------------------#
print(f"\n {sponsor} - {location}, {country_code}")
print(f" Download Bandwidth: {download_result:.2f} Mbps")
print(f"   Upload Bandwidth: {upload_result:.2f} Mbps")
print(f"     Latency (ping): {ping_result} ms")

input("\n Press Enter to exit")
```

Example run:

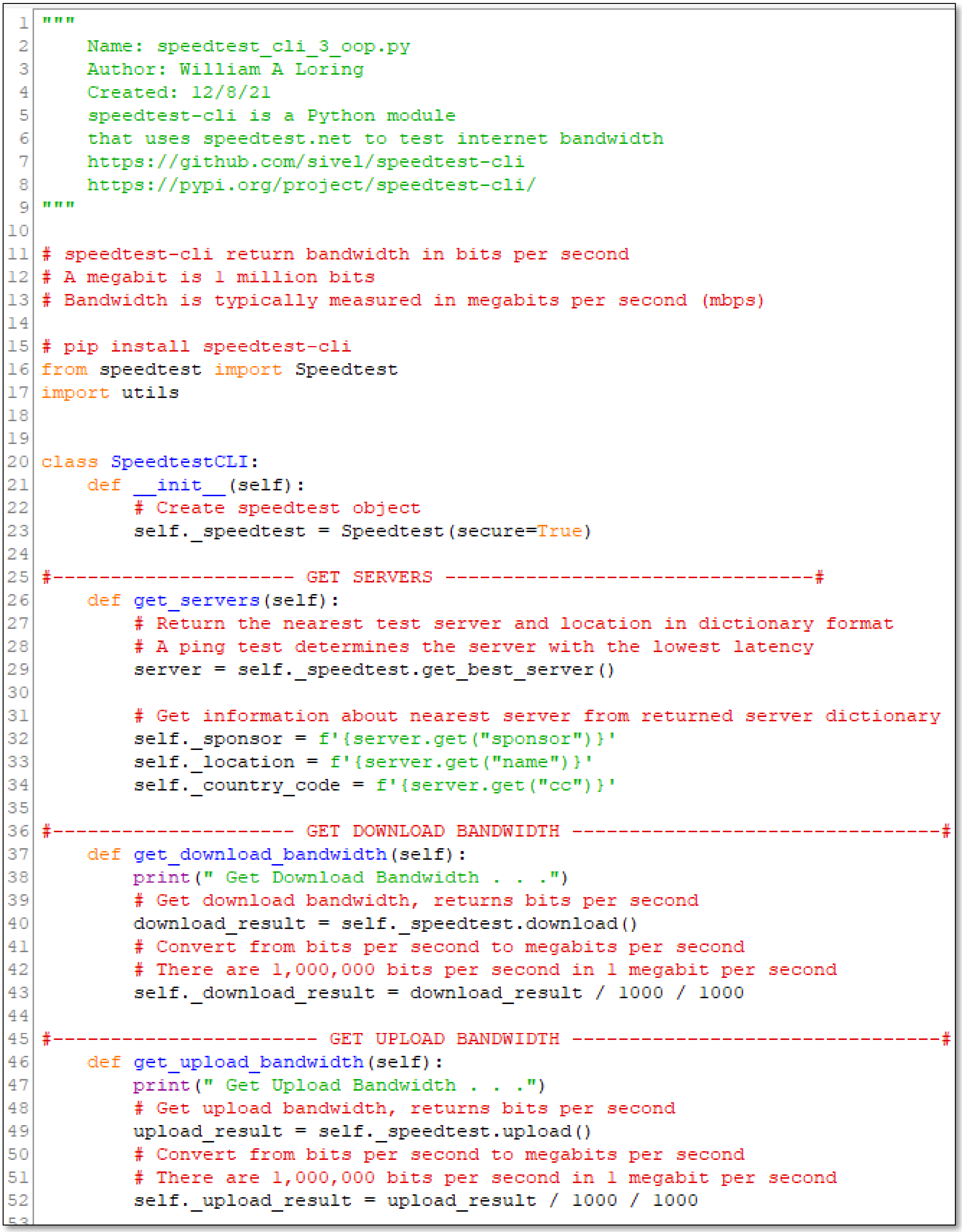


```
Start SpeedTest . . .
Get Download Bandwidth . . .
Get Upload Bandwidth . . .
Get Ping Latency . . .

Viaero Wireless - Fort Morgan, CO, US
Download Bandwidth: 565.36 Mbps
  Upload Bandwidth: 258.20 Mbps
    Latency (ping): 11.313 ms
```

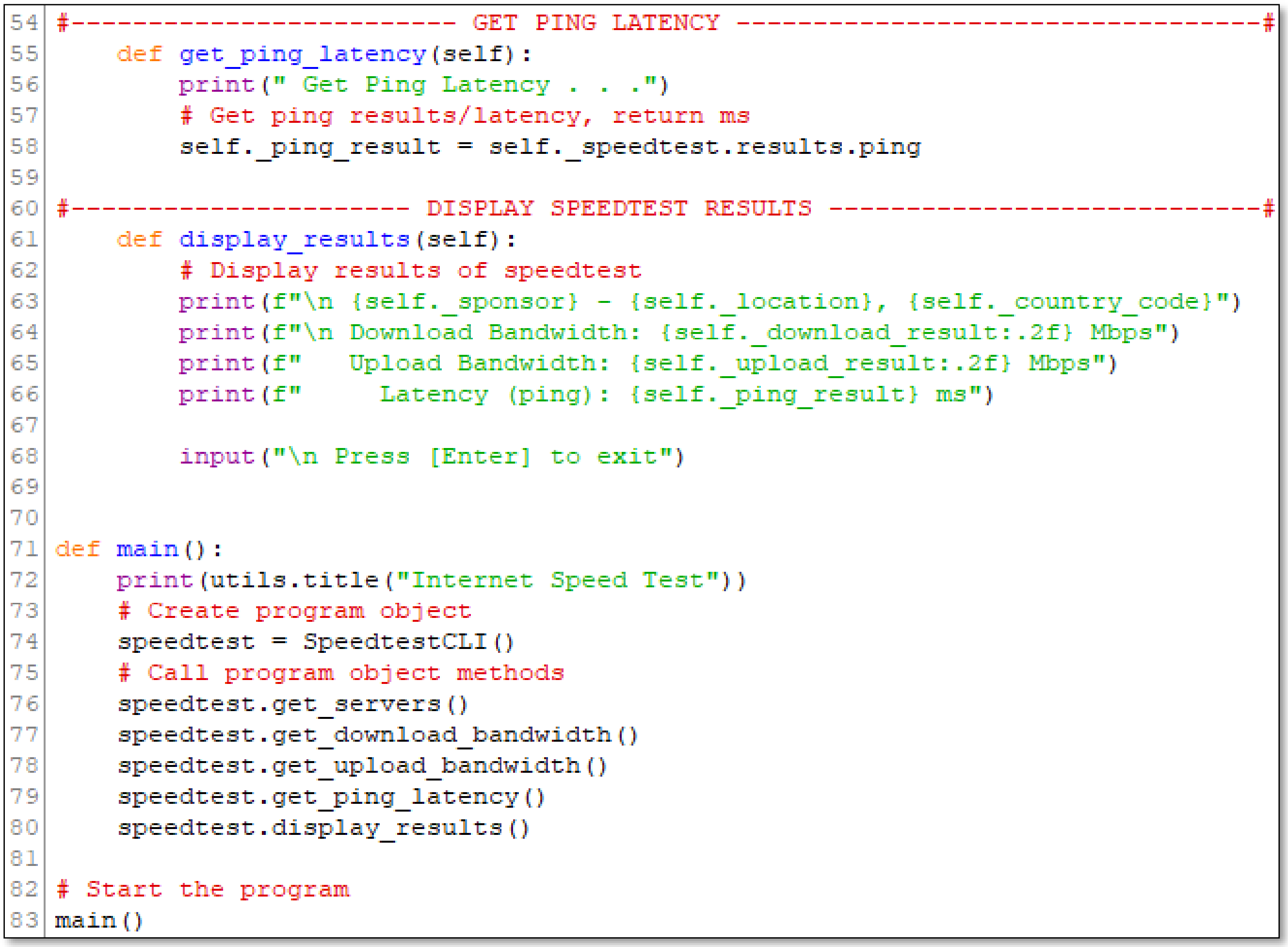
## Tutorial 4: Speedtest CLI OOP

This version converts the program to OOP. The code is pretty much the same, just in OOP format.

Save the previous program as **speedtest_cli_3.py**

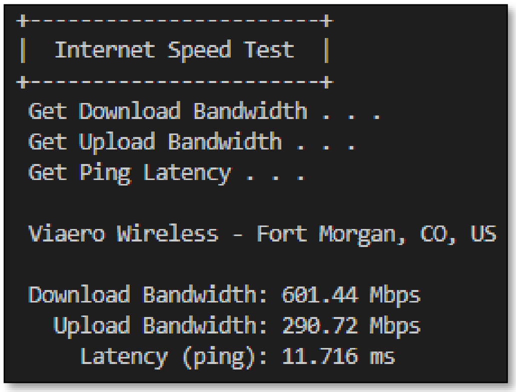


```python
"""
    Name: speedtest_cli_3_oop.py
    Author: William A Loring
    Created: 12/8/21
    speedtest-cli is a Python module
    that uses speedtest.net to test internet bandwidth
    https://github.com/sivel/speedtest-cli
    https://pypi.org/project/speedtest-cli/
"""

# speedtest-cli return bandwidth in bits per second
# A megabit is 1 million bits
# Bandwidth is typically measured in megabits per second (mbps)

# pip install speedtest-cli
from speedtest import Speedtest
import utils


class SpeedtestCLI:
    def __init__(self):
        # Create speedtest object
        self._speedtest = Speedtest(secure=True)

#--------------------- GET SERVERS --------------------------------#
    def get_servers(self):
        # Return the nearest test server and location in dictionary format
        # A ping test determines the server with the lowest latency
        server = self._speedtest.get_best_server()

        # Get information about nearest server from returned server dictionary
        self._sponsor = f'{server.get("sponsor")}'
        self._location = f'{server.get("name")}'
        self._country_code = f'{server.get("cc")}'

#--------------------- GET DOWNLOAD BANDWIDTH --------------------------------#
    def get_download_bandwidth(self):
        print(" Get Download Bandwidth . . .")
        # Get download bandwidth, returns bits per second
        download_result = self._speedtest.download()
        # Convert from bits per second to megabits per second
        # There are 1,000,000 bits per second in 1 megabit per second
        self._download_result = download_result / 1000 / 1000

#----------------------- GET UPLOAD BANDWIDTH --------------------------------#
    def get_upload_bandwidth(self):
        print(" Get Upload Bandwidth . . .")
        # Get upload bandwidth, returns bits per second
        upload_result = self._speedtest.upload()
        # Convert from bits per second to megabits per second
        # There are 1,000,000 bits per second in 1 megabit per second
        self._upload_result = upload_result / 1000 / 1000
```



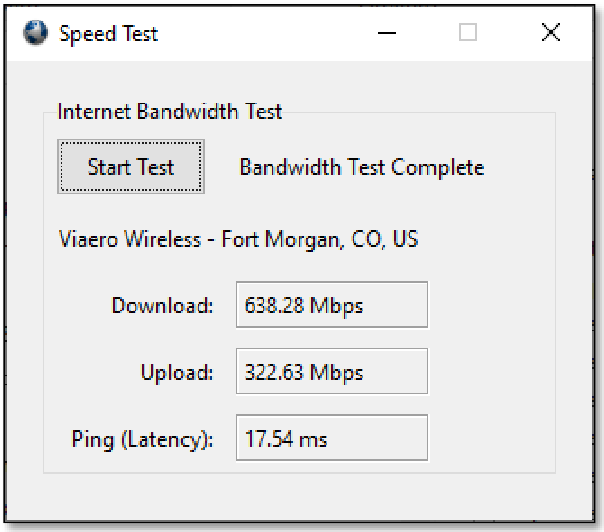
```python
#------------------------- GET PING LATENCY ----------------------------------#
    def get_ping_latency(self):
        print(" Get Ping Latency . . .")
        # Get ping results/latency, return ms
        self._ping_result = self._speedtest.results.ping

#---------------------- DISPLAY SPEEDTEST RESULTS ----------------------------#
    def display_results(self):
        # Display results of speedtest
        print(f"\n {self._sponsor} - {self._location}, {self._country_code}")
        print(f"\n Download Bandwidth: {self._download_result:.2f} Mbps")
        print(f"   Upload Bandwidth: {self._upload_result:.2f} Mbps")
        print(f"     Latency (ping): {self._ping_result} ms")

        input("\n Press [Enter] to exit")


def main():
    print(utils.title("Internet Speed Test"))
    # Create program object
    speedtest = SpeedtestCLI()
    # Call program object methods
    speedtest.get_servers()
    speedtest.get_download_bandwidth()
    speedtest.get_upload_bandwidth()
    speedtest.get_ping_latency()
    speedtest.display_results()

# Start the program
main()
```

Example run:



```
+-----------------------+
|  Internet Speed Test  |
+-----------------------+
 Get Download Bandwidth . . .
 Get Upload Bandwidth . . .
 Get Ping Latency . . .

 Viaero Wireless - Fort Morgan, CO, US

 Download Bandwidth: 601.44 Mbps
   Upload Bandwidth: 290.72 Mbps
     Latency (ping): 11.716 ms
```

## Tutorial 5: Speedtest GUI Tkinter



Bandwidth Test Complete

Viaero Wireless - Fort Morgan, CO, US

Download: 638.28 Mbps

Upload: 322.63 Mbps

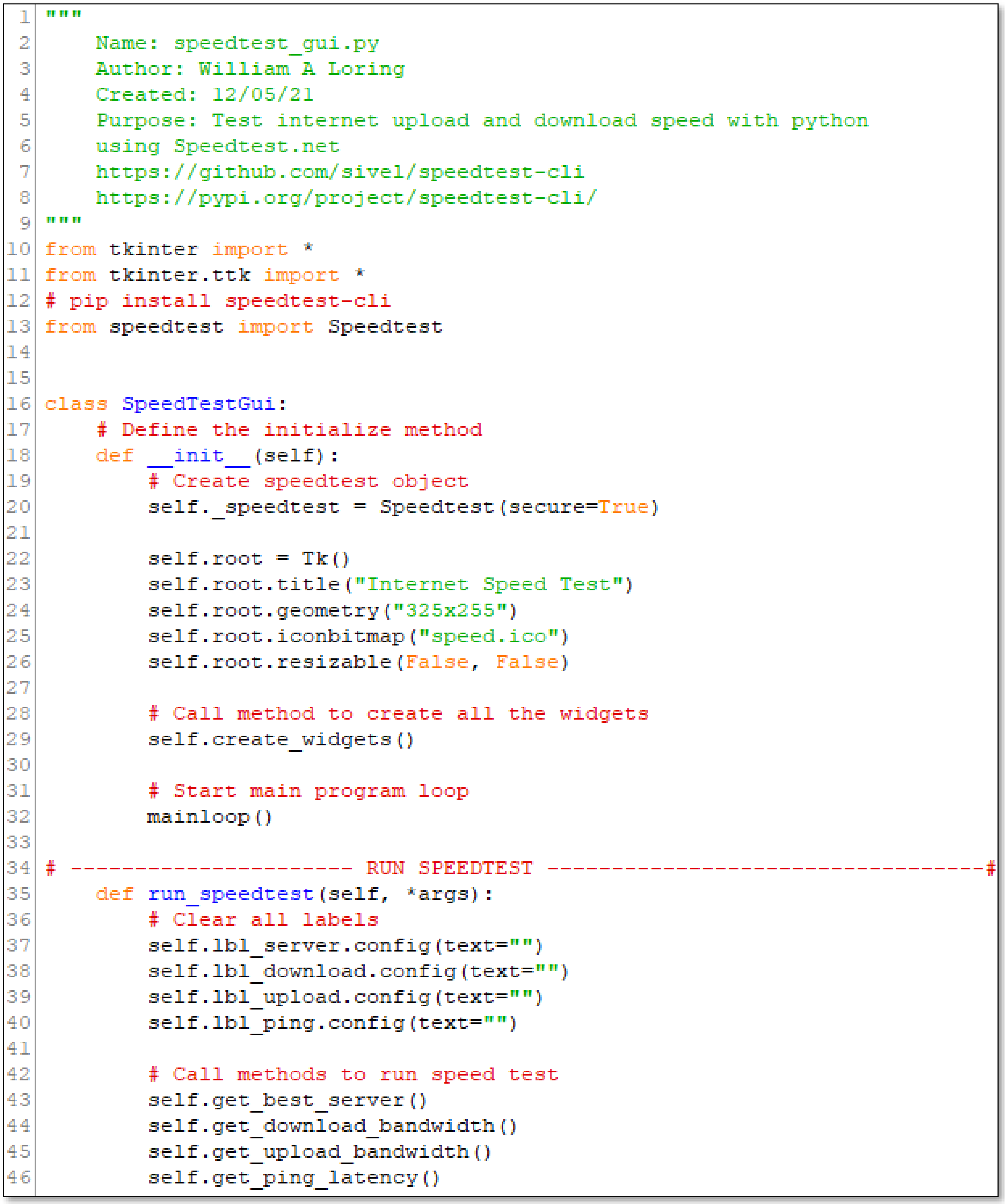
Ping (Latency): 17.54 ms

Sketch out the GUI before you start to build it. The example shown is built inside a single frame using the grid layout manager with two columns and multiple rows. It would be good practice to create your own UI.

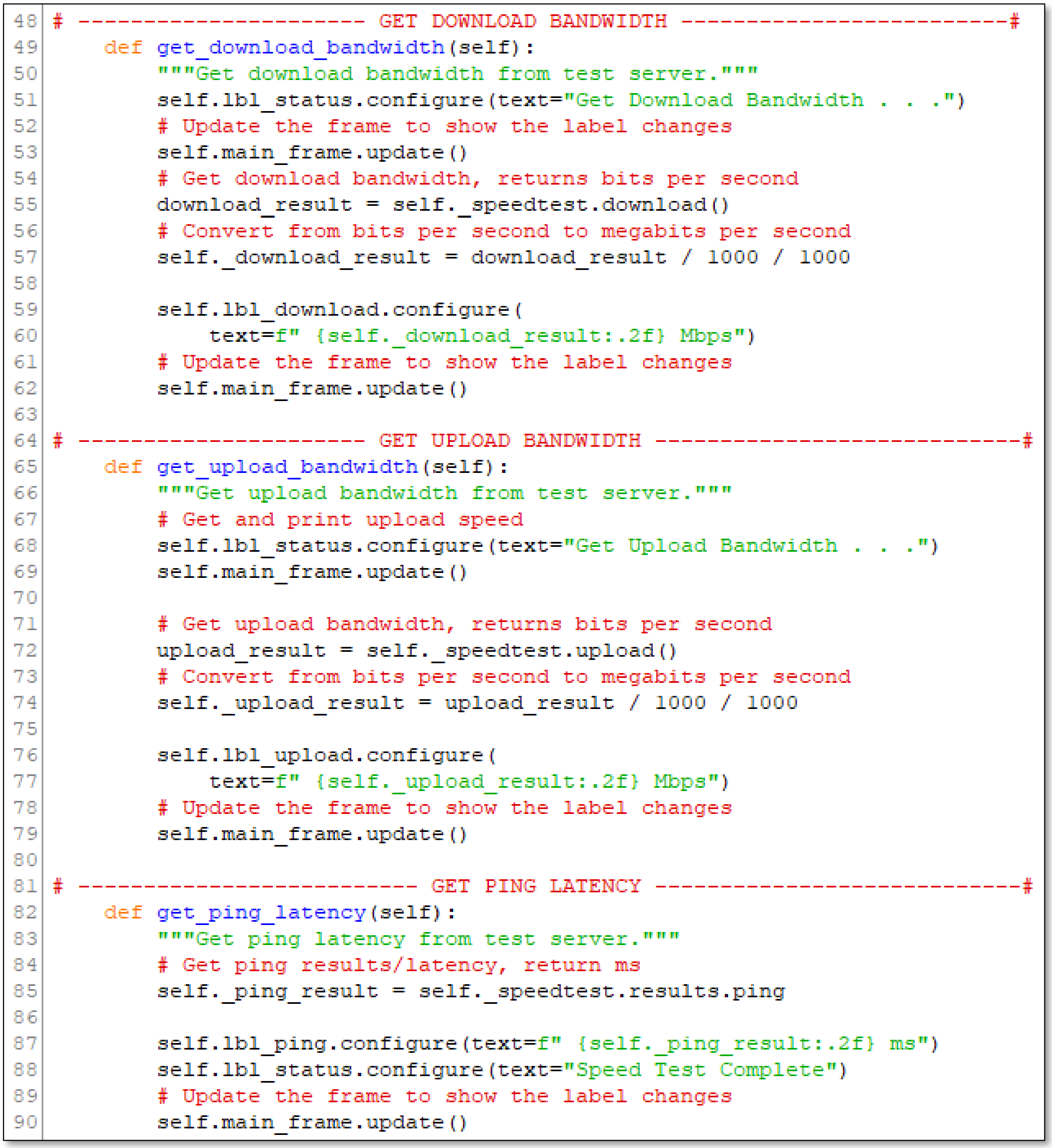
All the methods in the OOP CLI program will be modified to fit the GUI. All the logic will be the same. The changes will be to fit a Tkinter GUI program.

You will want to download an **internet.ico** file of some sort.

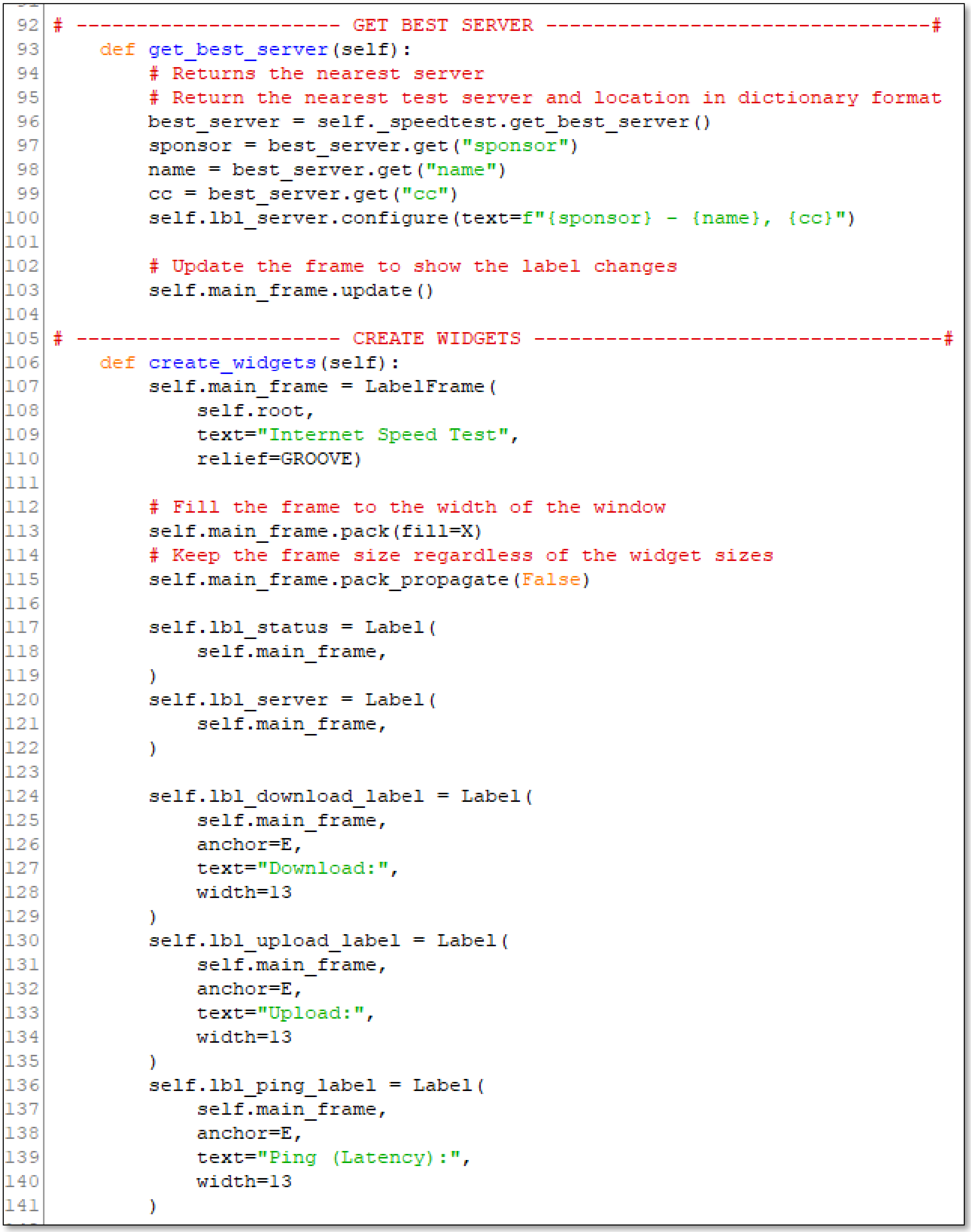
Save the last tutorial as **speedtest_gui.py**



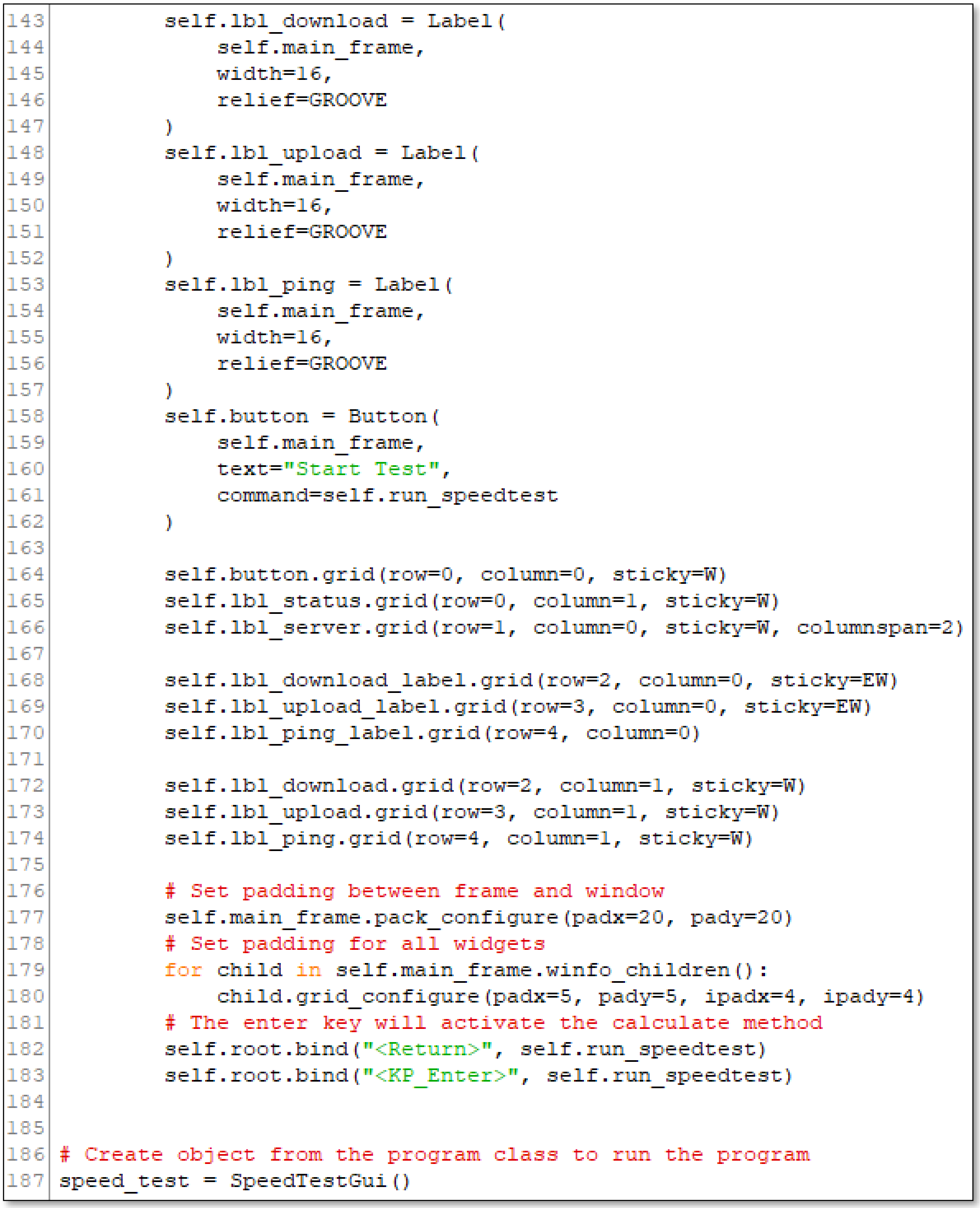
```python
"""
    Name: speedtest_gui.py
    Author: William A Loring
    Created: 12/05/21
    Purpose: Test internet upload and download speed with python
    using Speedtest.net
    https://github.com/sivel/speedtest-cli
    https://pypi.org/project/speedtest-cli/
"""
from tkinter import *
from tkinter.ttk import *
# pip install speedtest-cli
from speedtest import Speedtest


class SpeedTestGui:
    # Define the initialize method
    def __init__(self):
        # Create speedtest object
        self._speedtest = Speedtest(secure=True)

        self.root = Tk()
        self.root.title("Internet Speed Test")
        self.root.geometry("325x255")
        self.root.iconbitmap("speed.ico")
        self.root.resizable(False, False)

        # Call method to create all the widgets
        self.create_widgets()

        # Start main program loop
        mainloop()

# ---------------------- RUN SPEEDTEST ----------------------------------#
    def run_speedtest(self, *args):
        # Clear all labels
        self.lbl_server.config(text="")
        self.lbl_download.config(text="")
        self.lbl_upload.config(text="")
        self.lbl_ping.config(text="")

        # Call methods to run speed test
        self.get_best_server()
        self.get_download_bandwidth()
        self.get_upload_bandwidth()
        self.get_ping_latency()
```



```
# ---------------------- GET DOWNLOAD BANDWIDTH -------------------------#
    def get_download_bandwidth(self):
        """Get download bandwidth from test server."""
        self.lbl_status.configure(text="Get Download Bandwidth . . .")
        # Update the frame to show the label changes
        self.main_frame.update()
        # Get download bandwidth, returns bits per second
        download_result = self._speedtest.download()
        # Convert from bits per second to megabits per second
        self._download_result = download_result / 1000 / 1000

        self.lbl_download.configure(
            text=f" {self._download_result:.2f} Mbps")
        # Update the frame to show the label changes
        self.main_frame.update()

# ---------------------- GET UPLOAD BANDWIDTH ----------------------------#
    def get_upload_bandwidth(self):
        """Get upload bandwidth from test server."""
        # Get and print upload speed
        self.lbl_status.configure(text="Get Upload Bandwidth . . .")
        self.main_frame.update()

        # Get upload bandwidth, returns bits per second
        upload_result = self._speedtest.upload()
        # Convert from bits per second to megabits per second
        self._upload_result = upload_result / 1000 / 1000

        self.lbl_upload.configure(
            text=f" {self._upload_result:.2f} Mbps")
        # Update the frame to show the label changes
        self.main_frame.update()

# -------------------------- GET PING LATENCY ----------------------------#
    def get_ping_latency(self):
        """Get ping latency from test server."""
        # Get ping results/latency, return ms
        self._ping_result = self._speedtest.results.ping

        self.lbl_ping.configure(text=f" {self._ping_result:.2f} ms")
        self.lbl_status.configure(text="Speed Test Complete")
        # Update the frame to show the label changes
        self.main_frame.update()
```



```python
# ---------------------- GET BEST SERVER --------------------------------#
    def get_best_server(self):
        # Returns the nearest server
        # Return the nearest test server and location in dictionary format
        best_server = self._speedtest.get_best_server()
        sponsor = best_server.get("sponsor")
        name = best_server.get("name")
        cc = best_server.get("cc")
        self.lbl_server.configure(text=f"{sponsor} - {name}, {cc}")

        # Update the frame to show the label changes
        self.main_frame.update()

# ---------------------- CREATE WIDGETS ----------------------------------#
    def create_widgets(self):
        self.main_frame = LabelFrame(
            self.root,
            text="Internet Speed Test",
            relief=GROOVE)

        # Fill the frame to the width of the window
        self.main_frame.pack(fill=X)
        # Keep the frame size regardless of the widget sizes
        self.main_frame.pack_propagate(False)

        self.lbl_status = Label(
            self.main_frame,
        )
        self.lbl_server = Label(
            self.main_frame,
        )

        self.lbl_download_label = Label(
            self.main_frame,
            anchor=E,
            text="Download:",
            width=13
        )
        self.lbl_upload_label = Label(
            self.main_frame,
            anchor=E,
            text="Upload:",
            width=13
        )
        self.lbl_ping_label = Label(
            self.main_frame,
            anchor=E,
            text="Ping (Latency):",
            width=13
        )
```



```
        self.lbl_download = Label(
            self.main_frame,
            width=16,
            relief=GROOVE
        )
        self.lbl_upload = Label(
            self.main_frame,
            width=16,
            relief=GROOVE
        )
        self.lbl_ping = Label(
            self.main_frame,
            width=16,
            relief=GROOVE
        )
        self.button = Button(
            self.main_frame,
            text="Start Test",
            command=self.run_speedtest
        )

        self.button.grid(row=0, column=0, sticky=W)
        self.lbl_status.grid(row=0, column=1, sticky=W)
        self.lbl_server.grid(row=1, column=0, sticky=W, columnspan=2)

        self.lbl_download_label.grid(row=2, column=0, sticky=EW)
        self.lbl_upload_label.grid(row=3, column=0, sticky=EW)
        self.lbl_ping_label.grid(row=4, column=0)

        self.lbl_download.grid(row=2, column=1, sticky=W)
        self.lbl_upload.grid(row=3, column=1, sticky=W)
        self.lbl_ping.grid(row=4, column=1, sticky=W)

        # Set padding between frame and window
        self.main_frame.pack_configure(padx=20, pady=20)
        # Set padding for all widgets
        for child in self.main_frame.winfo_children():
            child.grid_configure(padx=5, pady=5, ipadx=4, ipady=4)
        # The enter key will activate the calculate method
        self.root.bind("<Return>", self.run_speedtest)
        self.root.bind("<KP_Enter>", self.run_speedtest)


# Create object from the program class to run the program
speed_test = SpeedTestGui()
```

## Assignment Submission

1. Attach all program files.
2. Attach a screenshot of each successful program run.

3. Submit the assignment in Blackboard.