# **Scout Data-Agent-focused Coding Exercise**

## ***Rohith Kumar Kaza***

**Exercise:**

Create a console application that creates a global Windows OS mouse hook, to collect the following information on every click done across any application running and print the following info on the console:

* Mouse coordinates in pixel
* Exe of the application running behind the cursor

**Solution:**

Programming Language used**:** Python

**Code:**

# These modules provide access to various Windows API functions and constants.

import win32api

import win32gui

import win32process

import win32con

# This module provides Python wrappers for Windows types.

import pywintypes

# This module from the pynput library enables listening to mouse events.

from pynput import mouse

# This function retrieves the executable name of the foreground window.

def get\_foreground\_window\_exe\_name():

    hwnd = win32gui.GetForegroundWindow()

    \_, pid = win32process.GetWindowThreadProcessId(hwnd)

    try:

        process\_handle = win32api.OpenProcess(

            win32con.PROCESS\_QUERY\_INFORMATION | win32con.PROCESS\_VM\_READ, False, pid

        )

        exe\_name = win32process.GetModuleFileNameEx(process\_handle, 0)

    # If any errors occur during this process, it returns "Unknown" as the executable name.

    except (win32api.error, pywintypes.error):

        exe\_name = "Unknown"

    return exe\_name

# This function is the callback for mouse events.

def on\_click(x, y, button, pressed):

    if pressed:

        exe\_name = get\_foreground\_window\_exe\_name()

        print(f"{exe\_name}: {{X={x}, Y={y}}}")

# This function sets up and starts the mouse listener.

def start\_mouse\_listener():

    listener = mouse.Listener(on\_click=on\_click)

    listener.start()

    try:

        while True:

            pass

    except KeyboardInterrupt:

        listener.stop()

if \_\_name\_\_ == "\_\_main\_\_":

    start\_mouse\_listener()

**Description:**

**1. Import Statements:**

* `win32api`, `win32gui`, `win32process`, `win32con`: These modules provide access to various Windows API functions and constants.
* `pywintypes`: This module provides Python wrappers for Windows types.
* `pynput.mouse`: This module from the `pynput` library enables listening to mouse events.

**2. `get\_foreground\_window\_exe\_name()` Function:**

* This function retrieves the executable name of the foreground window.
* It uses the `win32gui.GetForegroundWindow()` function to get the handle of the foreground window.
* Then it retrieves the process ID (`pid`) of the foreground window using `win32process.GetWindowThreadProcessId()`.
* The `win32api.OpenProcess()` function is used to open a handle to the process with specific access rights.
* Finally, it calls `win32process.GetModuleFileNameEx()` to get the executable name associated with the process.
* If any errors occur during this process, it returns "Unknown" as the executable name.

**3. `on\_click()` Function:**

* This function is the callback for mouse events.
* It is triggered whenever a mouse button is clicked (pressed or released).
* It calls `get\_foreground\_window\_exe\_name()` to get the executable name of the foreground window.
* Then it prints the executable name, mouse coordinates (`x` and `y`), and the status of the button (pressed or released).

**4. `start\_mouse\_listener()` Function:**

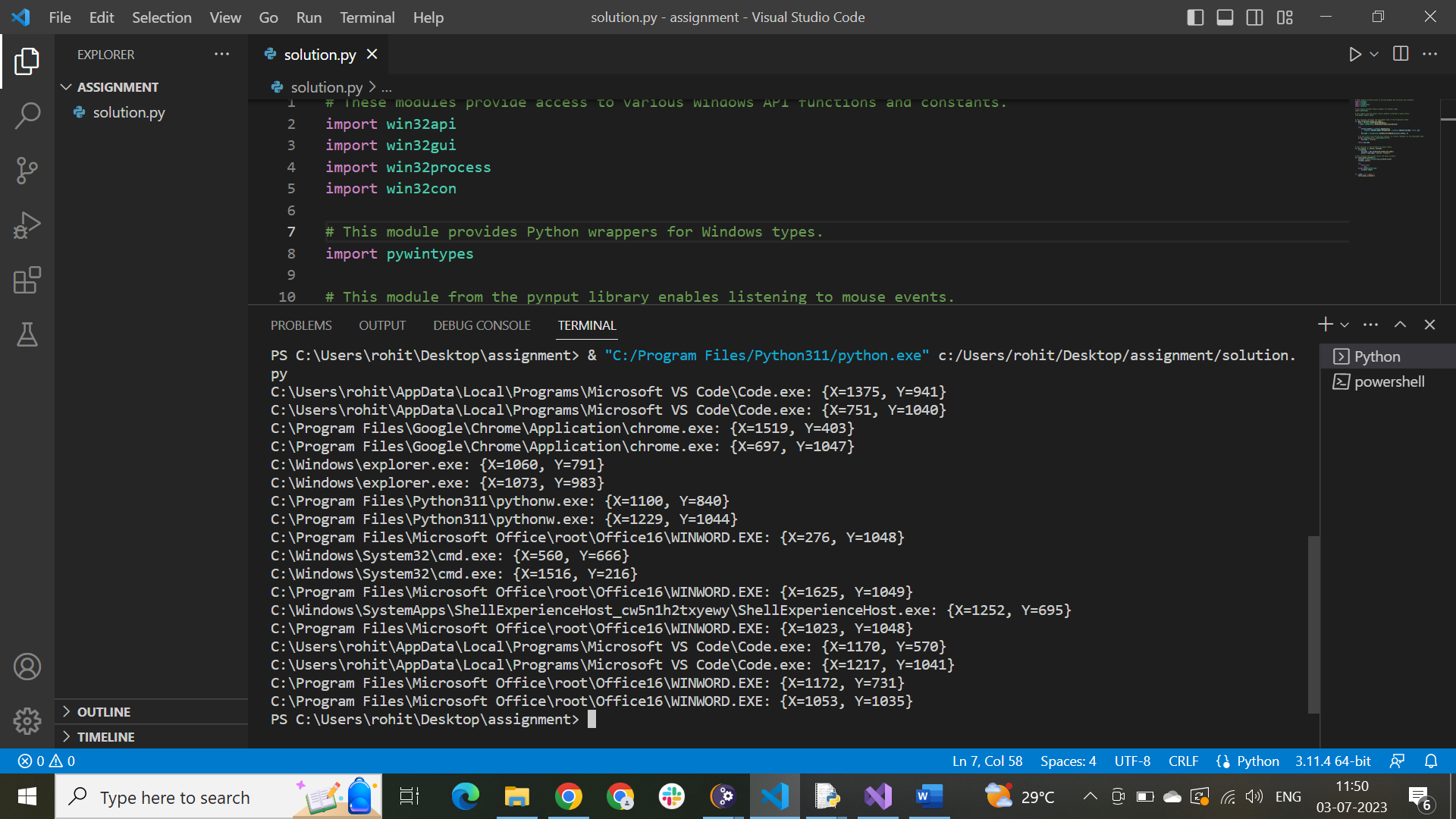
* This function sets up and starts the mouse listener.
* It creates a `mouse.Listener` object, passing the `on\_click` function as the callback.
* The listener is started by calling its `start()` method.
* It uses a `while True` loop to keep the script running until it is interrupted by a keyboard interrupt (`KeyboardInterrupt` exception).
* When interrupted, it stops the listener by calling its `stop()` method.

**5. `\_\_name\_\_ == "\_\_main\_\_"` Block:**

* This block ensures that the code inside it is only executed when the script is run directly, not when it is imported as a module.
* It calls the `start\_mouse\_listener()` function to initiate the mouse listener and start capturing mouse events.

Results

**Results:**



**Output:**

C:\Users\rohit\AppData\Local\Programs\Microsoft VS Code\Code.exe: {X=1375, Y=941}

C:\Users\rohit\AppData\Local\Programs\Microsoft VS Code\Code.exe: {X=751, Y=1040}

C:\Program Files\Google\Chrome\Application\chrome.exe: {X=1519, Y=403}

C:\Program Files\Google\Chrome\Application\chrome.exe: {X=697, Y=1047}

C:\Windows\explorer.exe: {X=1060, Y=791}

C:\Windows\explorer.exe: {X=1073, Y=983}

C:\Program Files\Python311\pythonw.exe: {X=1100, Y=840}

C:\Program Files\Python311\pythonw.exe: {X=1229, Y=1044}

C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE: {X=276, Y=1048}

C:\Windows\System32\cmd.exe: {X=560, Y=666}

C:\Windows\System32\cmd.exe: {X=1516, Y=216}

C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE: {X=1625, Y=1049}

C:\Windows\SystemApps\ShellExperienceHost\_cw5n1h2txyewy\ShellExperienceHost.exe: {X=1252, Y=695}

C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE: {X=1023, Y=1048}

C:\Users\rohit\AppData\Local\Programs\Microsoft VS Code\Code.exe: {X=1170, Y=570}