



```
1  import tkinter as tk
2
3  win =tk.Tk()
4
5  strData = tk.StringVar()
6
7  strData.set('Hello StringVar')
8
9  varData = strData.get()
10
11 print(varData)
```

PROBLEMS OUTPUT TERMINAL ... Code + - [ ] [X] ... ^ X

```
● PS D:\Nam_III\HK241\PyThonNangCao\TH\Lab_4> python -u "d:\Nam_III\HK241
  \PyThonNangCao\TH\Lab_4\bai_2.py"
  Hello StringVar
○ PS D:\Nam_III\HK241\PyThonNangCao\TH\Lab_4>
```

1.3\_ Viết chương trình python để in các giá trị mặc định IntVar, DoubleVar and BooleanVar như hình:

```
1  import tkinter as tk
2
3  win =tk.Tk()
4
5  strData = tk.StringVar()
6
7  strData.set('Hello StringVar')
8
9  varData = strData.get()
10
11 print(varData)
12
13 print(tk.IntVar())
14 print(tk.DoubleVar())
15 print(tk.BooleanVar())
```

PROBLEMS OUTPUT TERMINAL ... Code + - [ ] [X] ... ^ X

- PS D:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4> python -u "d:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4\bai\_3.py"  
Hello StringVar  
PY\_VAR1  
PY\_VAR2  
PY\_VAR3
- PS D:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4>

1.4\_ Viết chương trình python để in các giá trị mặc định `intData = tk.IntVar()` và breakpoint như hình:

```
1 import tkinter as tk
2
3 win =tk.Tk()
4
5 intData = tk.IntVar()
6 print(intData)
7 print(intData.get())
8
9 print()
```

PROBLEMS OUTPUT TERMINAL ... Code + - [ ] [ ] ... ^ X

● PS D:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4> python -u "d:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4\bai\_1.4.py"

PY\_VAR0  
0

**2. Lấy dữ liệu từ widget:** Viết chương trình python để dùng phương thức get() để lấy giá trị như hình:

```
210
211 strData = spin.get()
212 print("SpinBox value: " + strData)
213
```

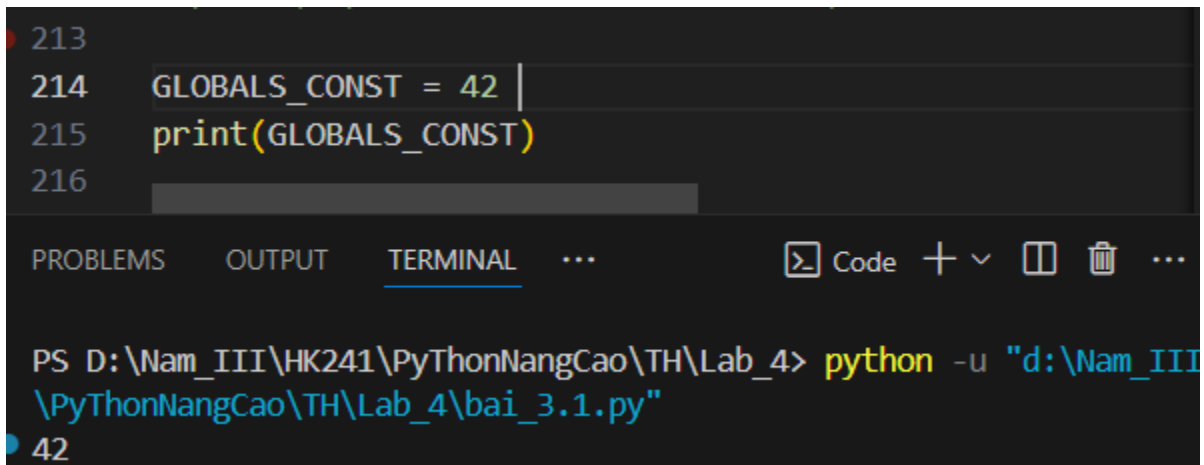
PROBLEMS OUTPUT TERMINAL ... Code + - [ ] [ ] ...

PS D:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4> python -u "d:\Nam\_III\PyThonNangCao\TH\Lab\_4\bai\_2.py"

SpinBox value: 1

### 3. Sử dụng biến toàn cục module-level:

3.1\_ Viết chương trình python để tạo biến toàn cục module-level và thêm hằng số như hình:



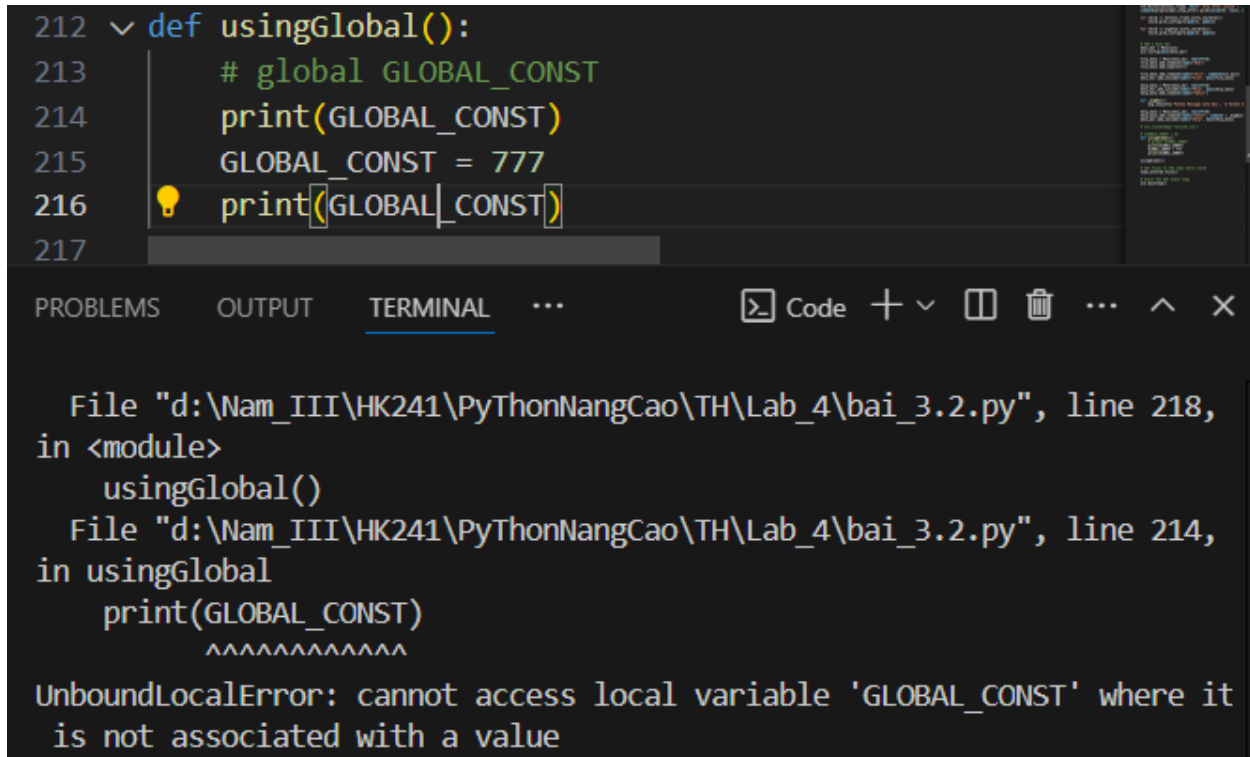
```
213
214 GLOBALS_CONST = 42
215 print(GLOBALS_CONST)
216
```

PROBLEMS OUTPUT TERMINAL ... Code + - ▢ 🗑 ...

PS D:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4> python -u "d:\Nam\_III\PyThonNangCao\TH\Lab\_4\bai\_3.1.py"

42

3.2\_ Viết chương trình python để in ra giá trị của biến toàn cục mà không sử dụng từ khóa toàn cục như hình:

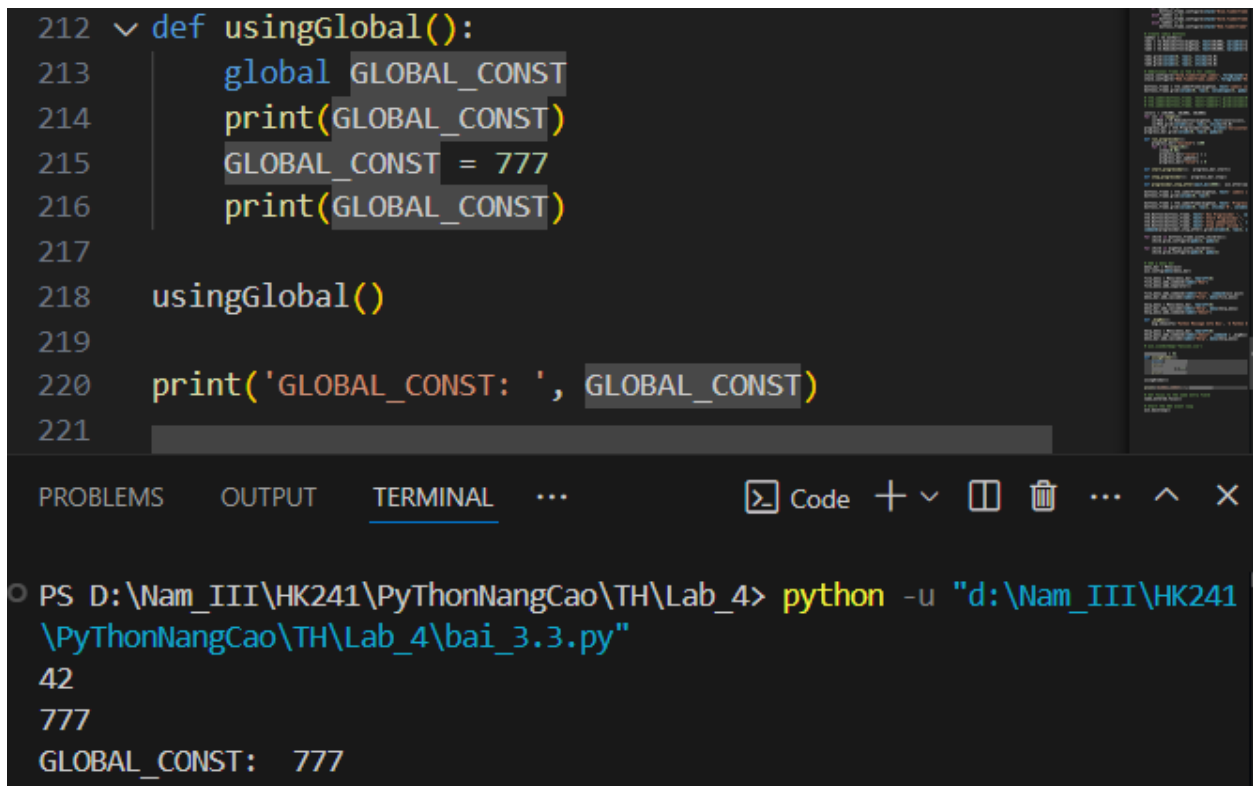


```
212 def usingGlobal():
213     # global GLOBAL_CONST
214     print(GLOBAL_CONST)
215     GLOBAL_CONST = 777
216     print(GLOBAL_CONST)
217
```

PROBLEMS OUTPUT TERMINAL ... Code + - ▢ 🗑 ... ^ X

File "d:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4\bai\_3.2.py", line 218, in <module>  
 usingGlobal()  
File "d:\Nam\_III\HK241\PyThonNangCao\TH\Lab\_4\bai\_3.2.py", line 214, in usingGlobal  
 print(GLOBAL\_CONST)  
 ^^^^^^^^^^^^^^^  
UnboundLocalError: cannot access local variable 'GLOBAL\_CONST' where it is not associated with a value

3.3\_ Viết chương trình python dùng hàm với biến toàn cục cũng là cục bộ như hình:



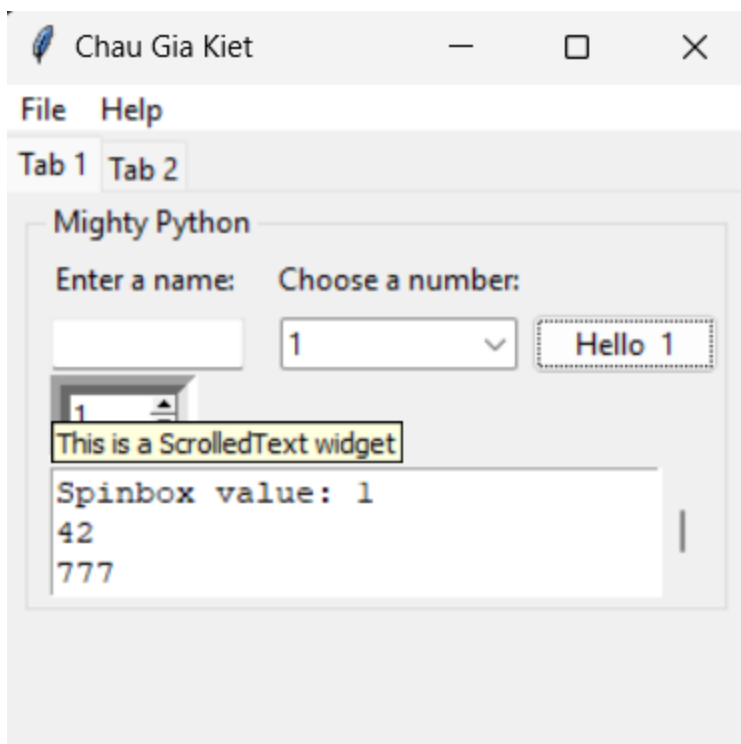
The image shows a screenshot of a Visual Studio Code editor window. The main editor area displays a Python script with the following code:

```
212 def usingGlobal():
213     global GLOBAL_CONST
214     print(GLOBAL_CONST)
215     GLOBAL_CONST = 777
216     print(GLOBAL_CONST)
217
218 usingGlobal()
219
220 print('GLOBAL_CONST: ', GLOBAL_CONST)
221
```

Below the code editor, the 'TERMINAL' tab is active, showing the command prompt and the output of the script:

```
PS D:\Nam_III\HK241\PyThonNangCao\TH\Lab_4> python -u "d:\Nam_III\HK241
PyThonNangCao\TH\Lab_4\bai_3.3.py"
42
777
GLOBAL_CONST: 777
```

**4. Cải tiến lại chương trình bằng cách tạo class (OOP):** Viết chương trình với 2 lớp OOP và ToolTip như hình:



**5. Tạo các thành phần GUI có thể sử dụng lại:** Viết chương trình sử dụng lại Tooltip cho các widget như hình:

