

Python First Steps

Hello Python world

Reading: Chapters 1&2 from Zelle text

Data processing

- Quantities of interest are digitized into data
- Data is organized in data structures
- Algorithms make use of data structures to specify data processing step by step
 - Described in pseudocode
- Algorithms converted to computer programs
 - Pseudocode translated into program by programmer
 - Program written in high-level programming language
- Program translated into machine code
 - Automatically, by special SW called compiler
- Machine code executed on computer
 - With the help of special SW called operating system

Data processing

- Quantities of interest are digitized into data
- Data is organized in data structures
- Algorithms make use of data structures to specify data processing step by step
 - Described in pseudocode
- Algorithms converted to computer programs
 - Pseudocode translated into program by programmer
 - Program written in high-level programming language
- Program translated into machine code
 - Automatically, by special SW called compiler
- Machine code executed on computer
 - With the help of special SW called operating system

Programming

- Algorithms converted to computer programs
 - Pseudocode translated into program by programmer
 - Program written in high-level programming language
 - We will be using Python
 - High-level programming languages are similar
 - Understanding of data, data structures, and algorithms (in pseudocode) ensures low learning curve for any high level programming language

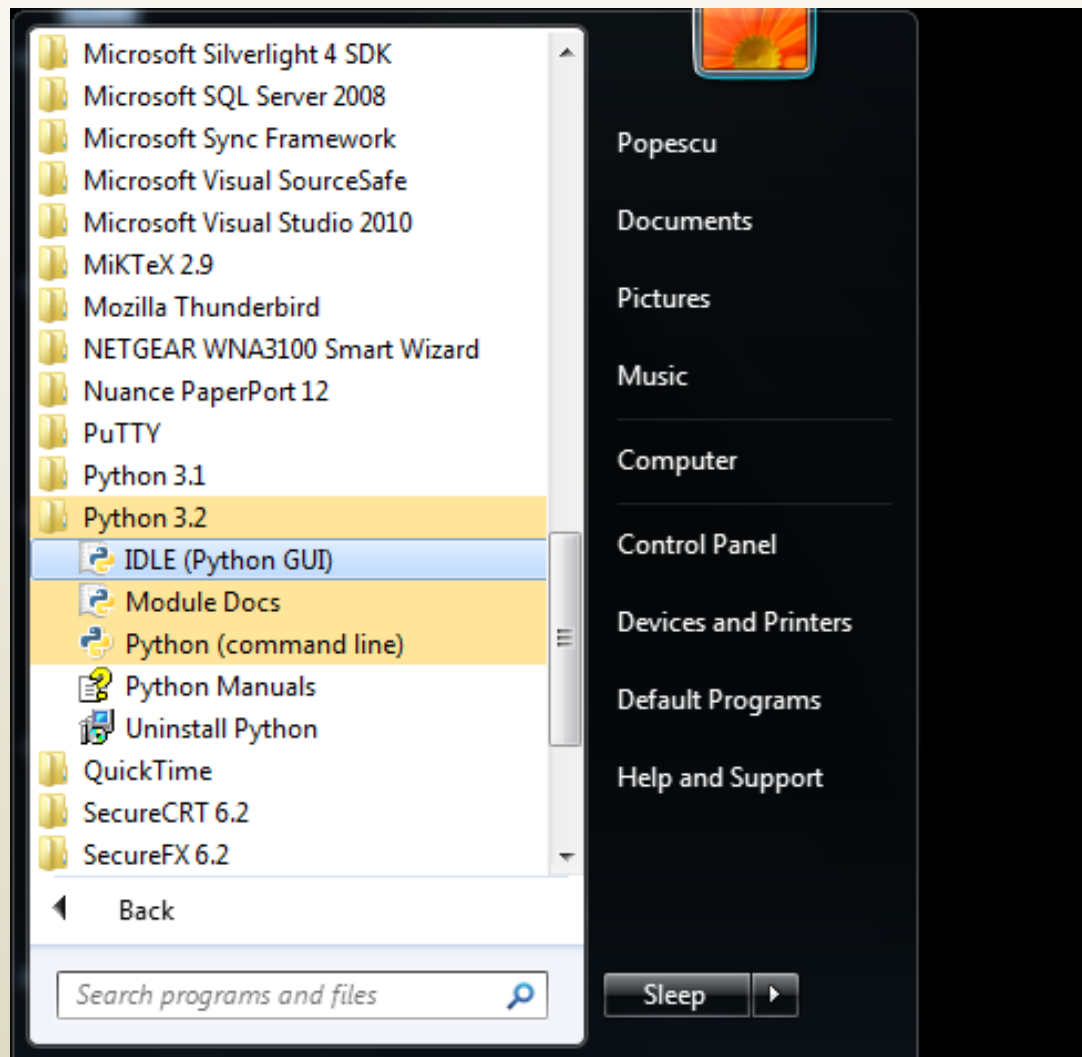
Programming in Python

- We will be programming in Python using IDLE
 - Integrated development environment
 - Essentially a Graphical User Interface (GUI) that makes Python easier to use
 - Allows writing, saving, executing, debugging Python programs

Example 1

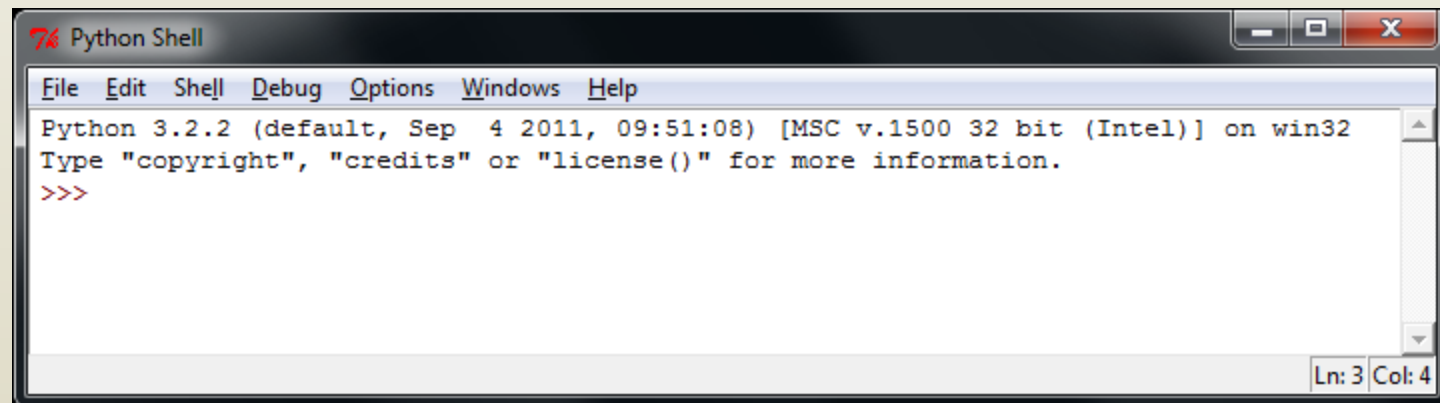
- Converting Celsius to Fahrenheit

1. Run IDLE



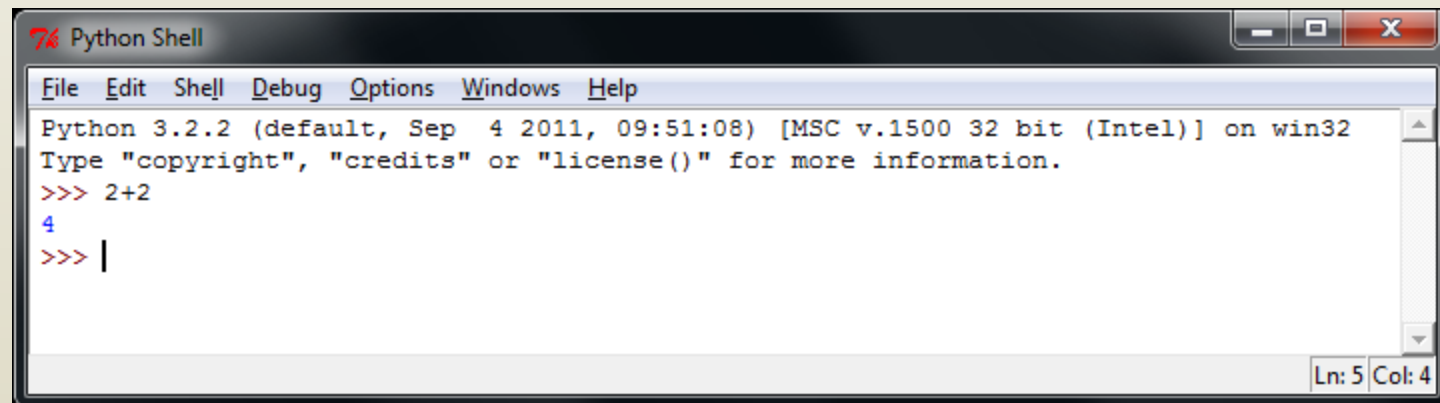
1. Run IDLE

- Python Shell window comes up
 - A prompt, i.e. “>>>”, awaits for user commands



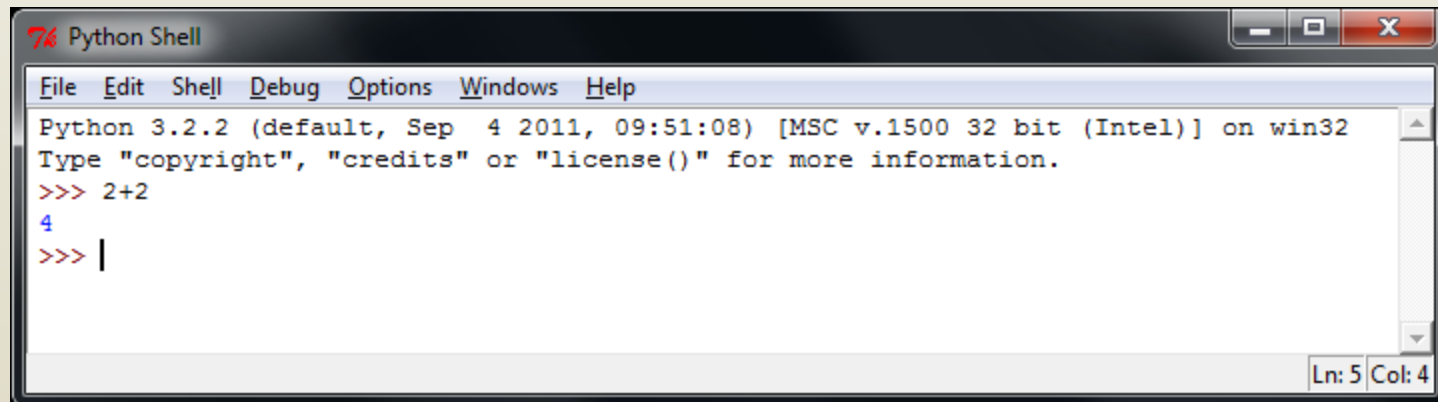
1. Run IDLE

- Python Shell window comes up
 - A prompt, i.e. “>>>”, awaits for user commands
 - For example one can use Python as a calculator



1. Run IDLE

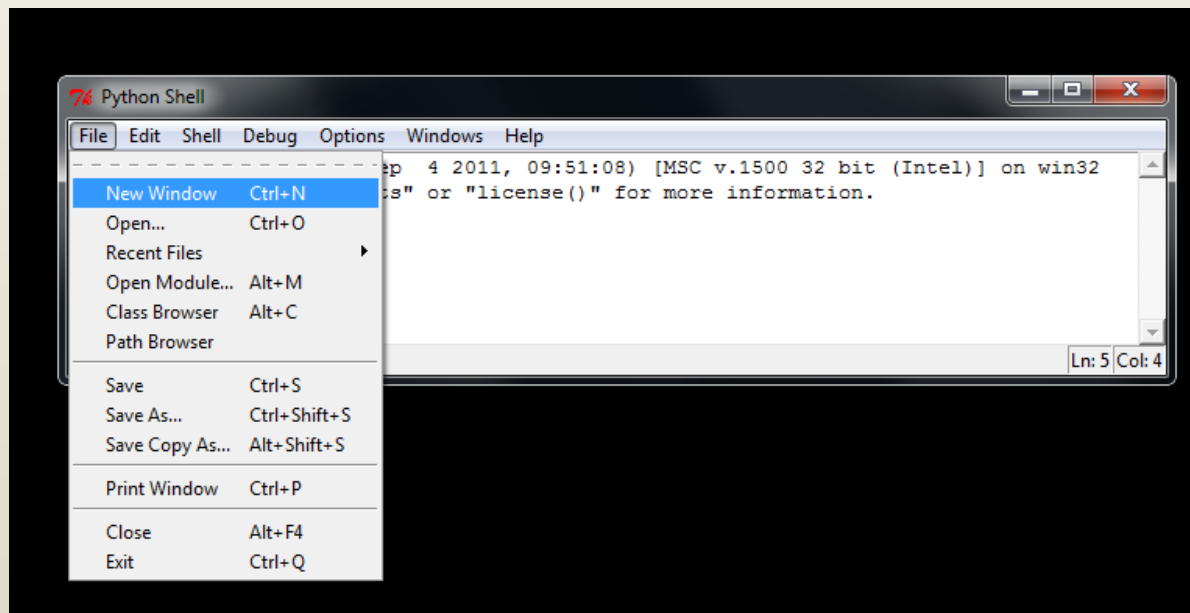
- Python Shell window comes up
 - A prompt, i.e. “>>>”, awaits for user commands
 - For example one can use Python as a calculator
 - We do not write our Python program in the shell window



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep  4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 2+2
4
>>> |
Ln: 5 Col: 4
```

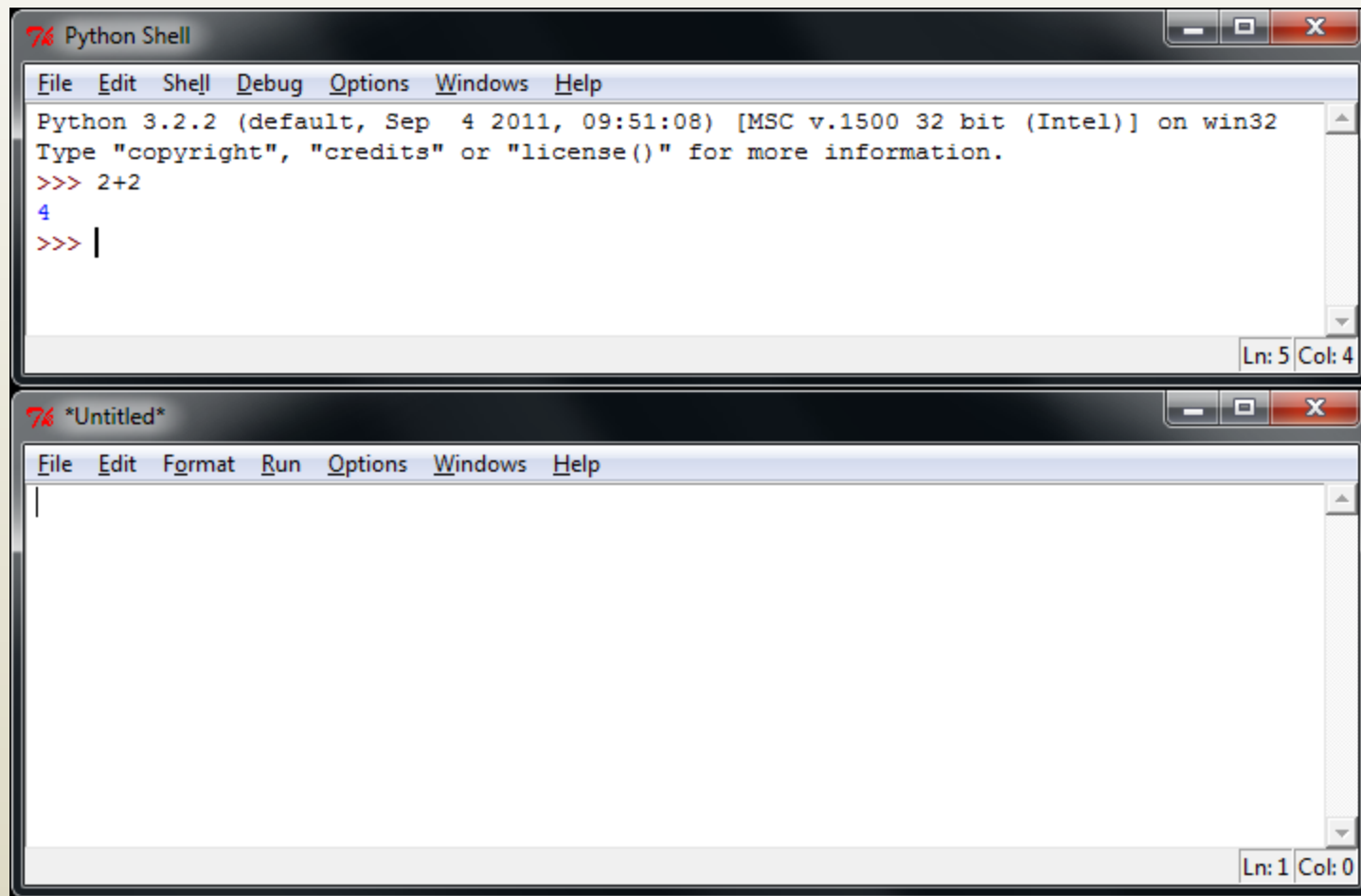
2. Make new source file

- Python Shell window comes up
 - We do not write program in shell window
 - Instead we open a source file in a new window (source file is text file where we write program)



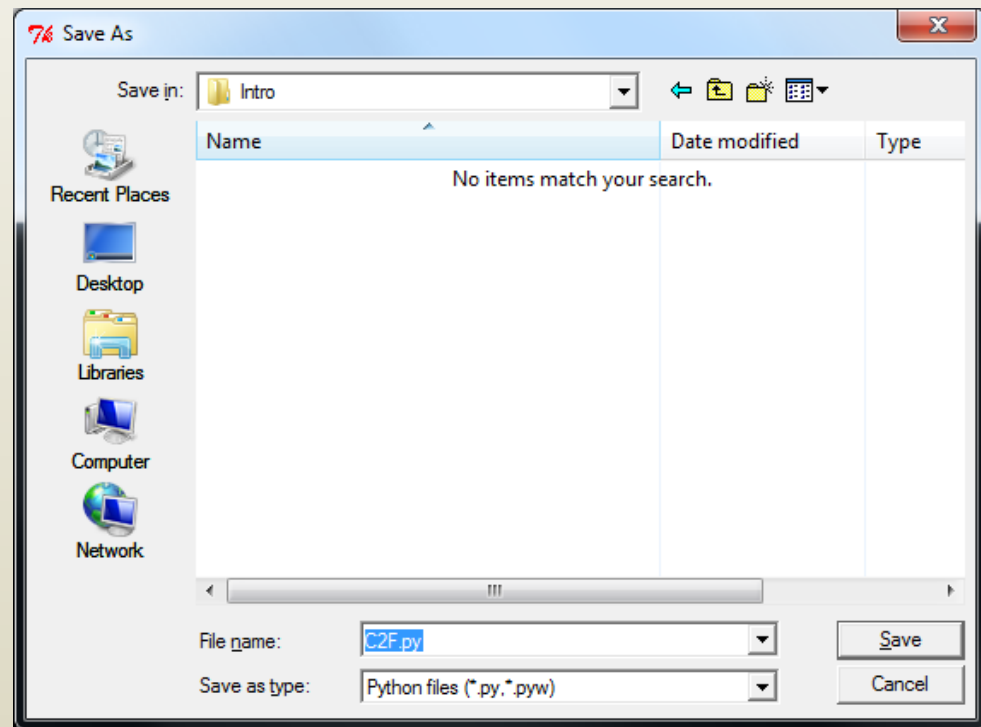
2. Make new source file

- Two windows: shell and source file



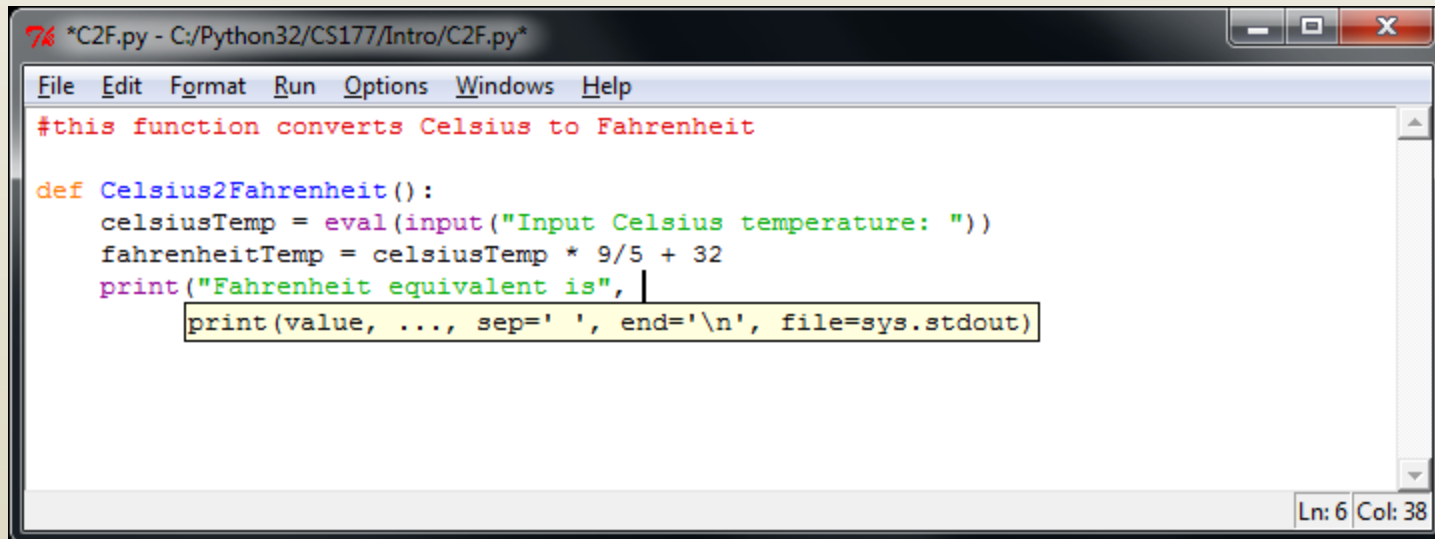
3. Save source file

- Choose suggestive name: C2F.py
 - Celsius to Fahrenheit
 - extension py from Python (same for all Python source files)



4. Write Python program

- IDLE helps (special editor that knows Python)
 - Syntactic highlighting
 - Red for comments
 - Green for text strings, etc
 - Syntax of what is currently typed



The screenshot shows the IDLE Python editor window. The title bar reads '*C2F.py - C:/Python32/CS177/Intro/C2F.py*'. The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code is as follows:

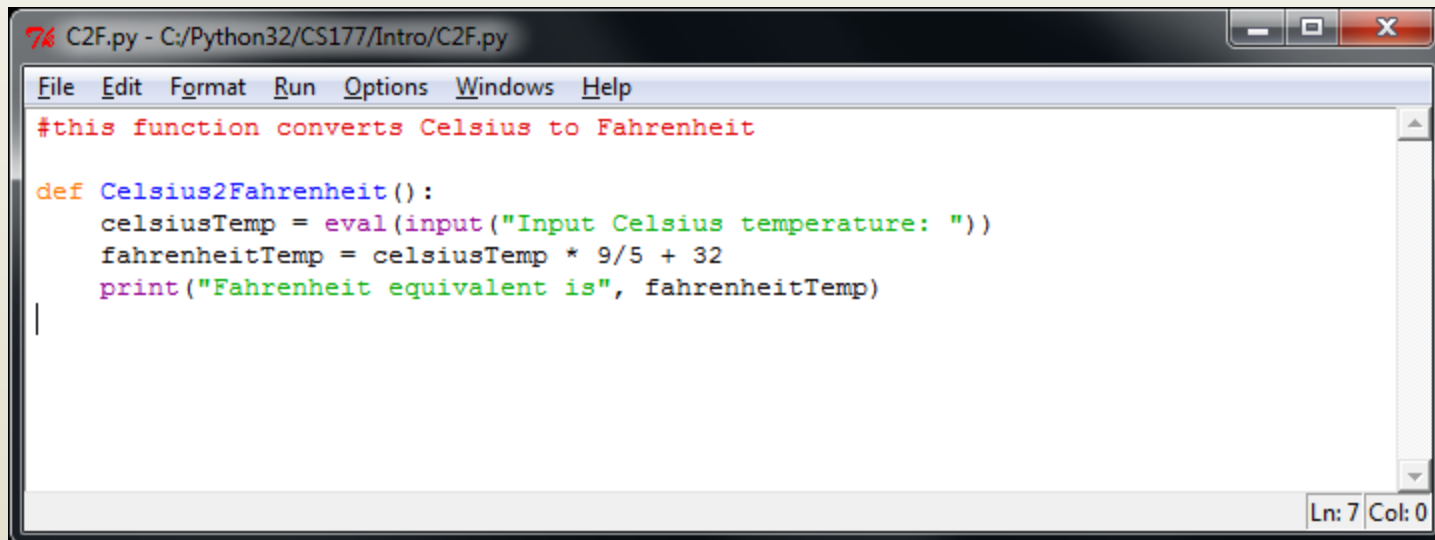
```
#this function converts Celsius to Fahrenheit

def Celsius2Fahrenheit():
    celsiusTemp = eval(input("Input Celsius temperature: "))
    fahrenheitTemp = celsiusTemp * 9/5 + 32
    print("Fahrenheit equivalent is", |
        print(value, ..., sep=' ', end='\n', file=sys.stdout))
```

The status bar at the bottom right indicates 'Ln: 6 Col: 38'.

4. Write Python program

- Save program (early and often)

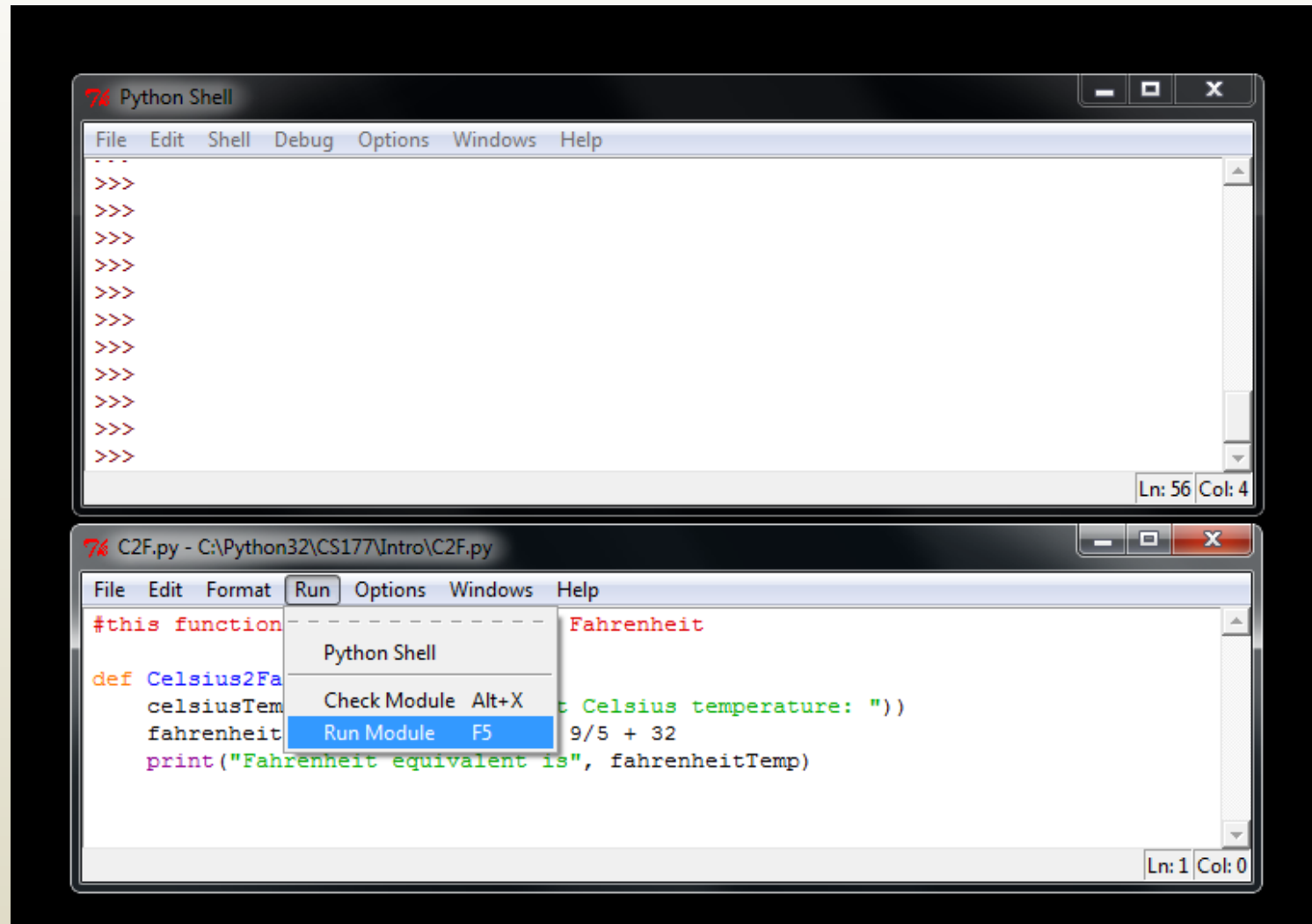
A screenshot of a Python IDE window titled 'C2F.py - C:/Python32/CS177/Intro/C2F.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main text area contains the following Python code:

```
#this function converts Celsius to Fahrenheit

def Celsius2Fahrenheit():
    celsiusTemp = eval(input("Input Celsius temperature: "))
    fahrenheitTemp = celsiusTemp * 9/5 + 32
    print("Fahrenheit equivalent is", fahrenheitTemp)
```

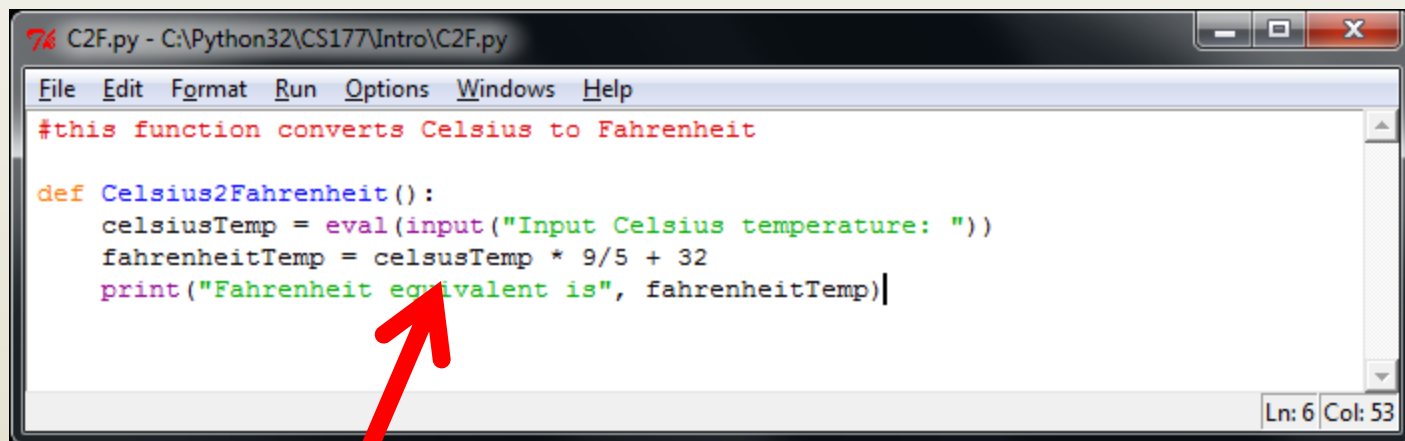
The code is color-coded: comments are red, function definitions are blue, and other code is black. The status bar at the bottom right shows 'Ln: 7 Col: 0'.

5. Run program



6. Remove syntax errors

- Let's say we misspelled celsiusTemp



The screenshot shows a Python IDE window titled "C2F.py - C:\Python32\CS177\Intro\C2F.py". The code inside the window is as follows:

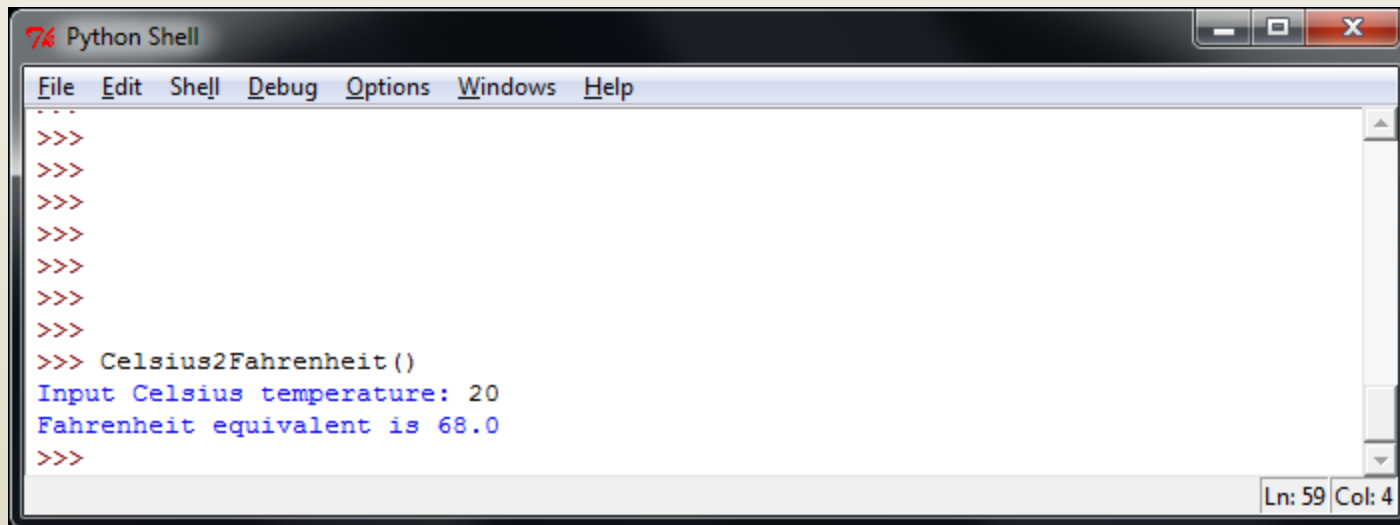
```
File Edit Format Run Options Windows Help
#this function converts Celsius to Fahrenheit

def Celsius2Fahrenheit():
    celsiusTemp = eval(input("Input Celsius temperature: "))
    fahrenheitTemp = celsiusTemp * 9/5 + 32
    print("Fahrenheit equivalent is", fahrenheitTemp)
```

A red arrow points to the word "equivalent" in the print statement, indicating a syntax error. The status bar at the bottom right shows "Ln: 6 Col: 53".

5. Run program

- Now the Python shell knows about our Celsius2Fahrenheit algorithm
- We can go in the shell window and run it



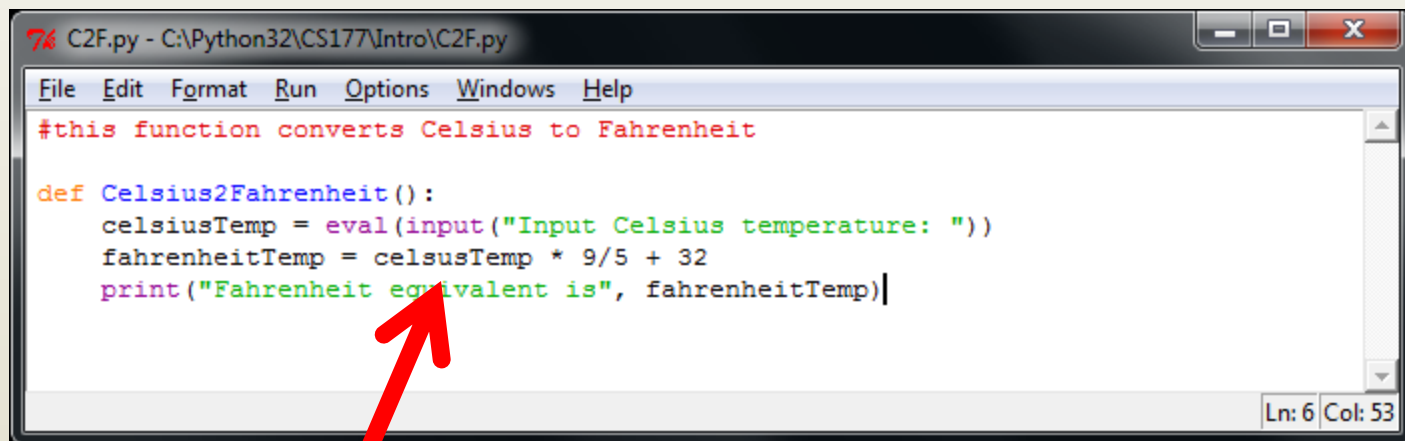
The screenshot shows a Python Shell window with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help) and a command prompt interface. The prompt shows several empty lines followed by the function call `Celsius2Fahrenheit()`. The output displays the input temperature and its Fahrenheit equivalent.

```
Python Shell
File Edit Shell Debug Options Windows Help
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> Celsius2Fahrenheit()
Input Celsius temperature: 20
Fahrenheit equivalent is 68.0
>>>
```

Ln: 59 Col: 4

6. Remove syntax errors

- Let's say we misspelled celsiusTemp



The screenshot shows a Python IDE window titled "C2F.py - C:\Python32\CS177\Intro\C2F.py". The code inside the window is as follows:

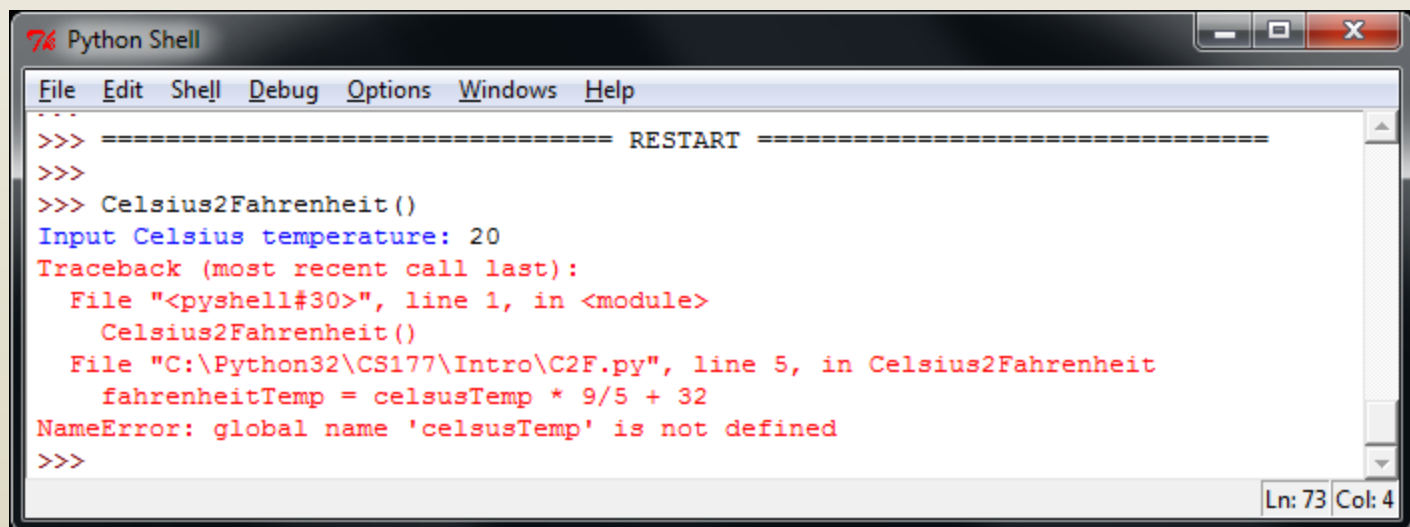
```
#this function converts Celsius to Fahrenheit

def Celsius2Fahrenheit():
    celsiusTemp = eval(input("Input Celsius temperature: "))
    fahrenheitTemp = celsiusTemp * 9/5 + 32
    print("Fahrenheit equivalent is", fahrenheitTemp)
```

A red arrow points to the word "equivalent" in the print statement, indicating a syntax error. The status bar at the bottom right shows "Ln: 6 Col: 53".

6. Remove syntax errors

- Let's say we misspelled celsiusTemp
- When execution gets to the line with the typo, we get an error message and execution stops
 - It tells us that in line 5 celsusTemp is not defined



The screenshot shows a Python Shell window with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help) and a command prompt interface. The user has entered several commands: a restart command, two empty prompts, and a function call `Celsius2Fahrenheit()`. The function prompts for 'Input Celsius temperature: 20'. A traceback error is displayed, indicating a `NameError: global name 'celsusTemp' is not defined` at line 5 of the file `C:\Python32\CS177\Intro\C2F.py`. The error message is in red text. The status bar at the bottom right shows 'Ln: 73 Col: 4'.

```
Python Shell
File Edit Shell Debug Options Windows Help
>>> ===== RESTART =====
>>>
>>> Celsius2Fahrenheit()
Input Celsius temperature: 20
Traceback (most recent call last):
  File "<pyshell#30>", line 1, in <module>
    Celsius2Fahrenheit()
  File "C:\Python32\CS177\Intro\C2F.py", line 5, in Celsius2Fahrenheit
    fahrenheitTemp = celsusTemp * 9/5 + 32
NameError: global name 'celsusTemp' is not defined
>>>
```

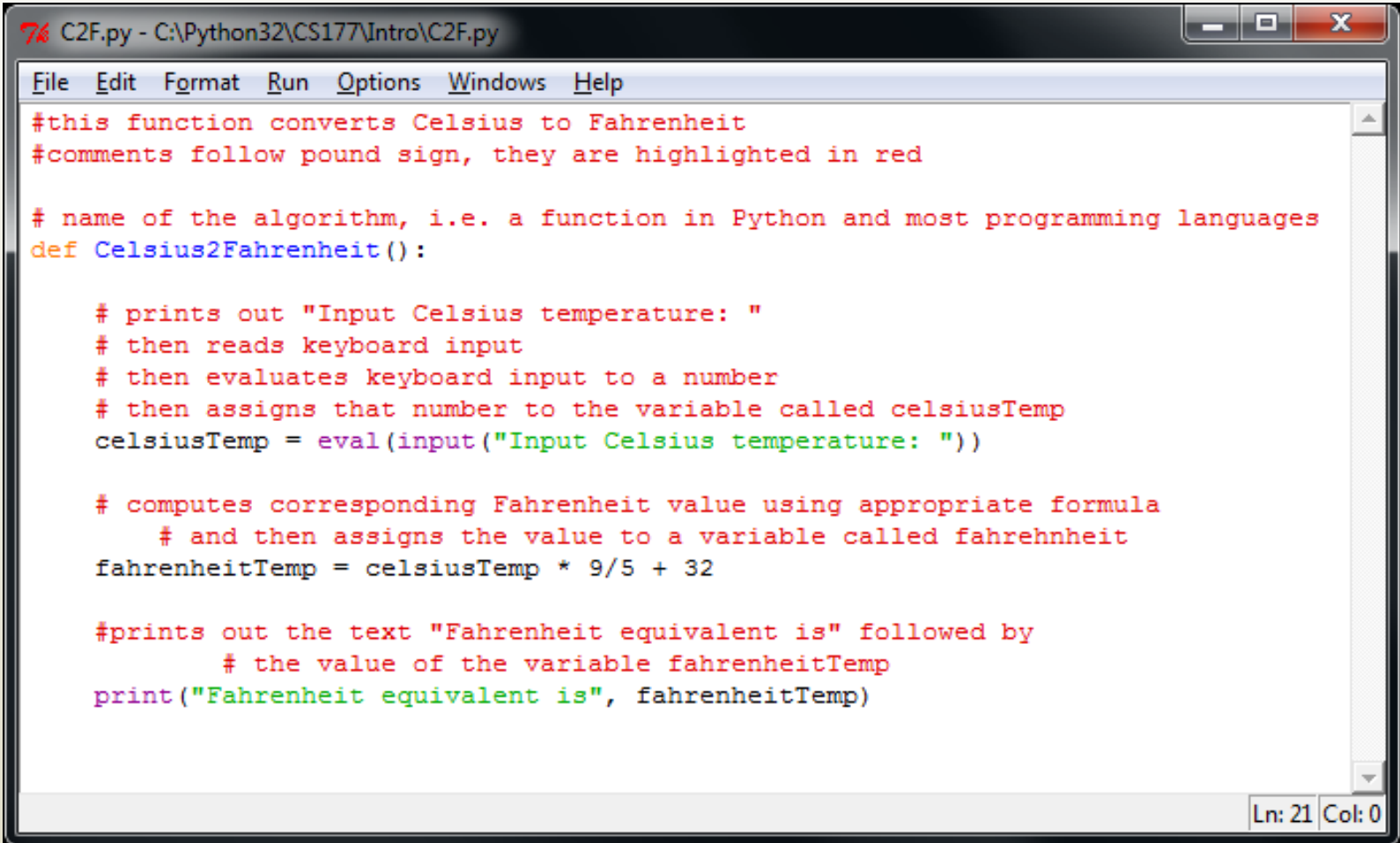
6. Remove syntax errors

- Notice that even though `celsusTemp` is very similar to `celsiusTemp`, Python interpreter
 - Does not try to guess that the programmer actually meant `celsiusTemp`
 - Does not fix the typo automatically
 - Does not even suggest that that's what the programmer meant

Interpreter vs compiler

- Python is interpreted as opposed to compiled
- Interpreter
 - Examines and executes one line at the time
 - Errors are found only once a line is executed
- Compiler
 - Examines entire program at once and translates into machine code
 - Finds syntax errors early

A closer look at the Python program



```
7% C2F.py - C:\Python32\CS177\Intro\C2F.py
File Edit Format Run Options Windows Help
#this function converts Celsius to Fahrenheit
#comments follow pound sign, they are highlighted in red

# name of the algorithm, i.e. a function in Python and most programming languages
def Celsius2Fahrenheit():

    # prints out "Input Celsius temperature: "
    # then reads keyboard input
    # then evaluates keyboard input to a number
    # then assigns that number to the variable called celsiusTemp
    celsiusTemp = eval(input("Input Celsius temperature: "))

    # computes corresponding Fahrenheit value using appropriate formula
    # and then assigns the value to a variable called fahrenheitTemp
    fahrenheitTemp = celsiusTemp * 9/5 + 32

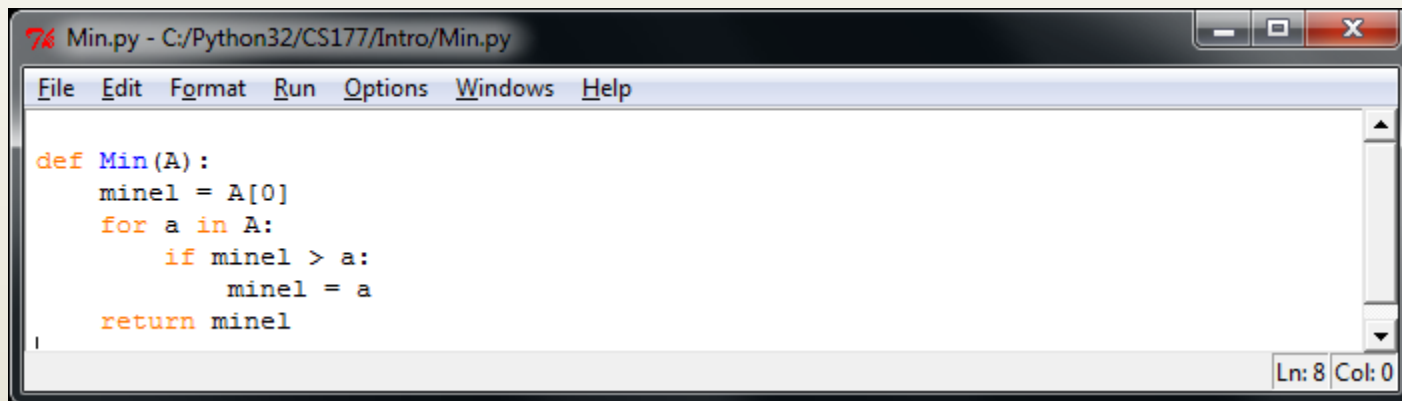
    #prints out the text "Fahrenheit equivalent is" followed by
    # the value of the variable fahrenheitTemp
    print("Fahrenheit equivalent is", fahrenheitTemp)
```

Ln: 21 Col: 0

iClicker question

- Which statement is true
 - A. IDLE is a SW environment that facilitates programming in Python
 - B. IDLE includes an editor that knows Python and provides syntactic highlighting
 - C. IDLE finds all programming errors
 - D. A, B, and C
 - E. A and B

Example 2: minimum in a 1-D array

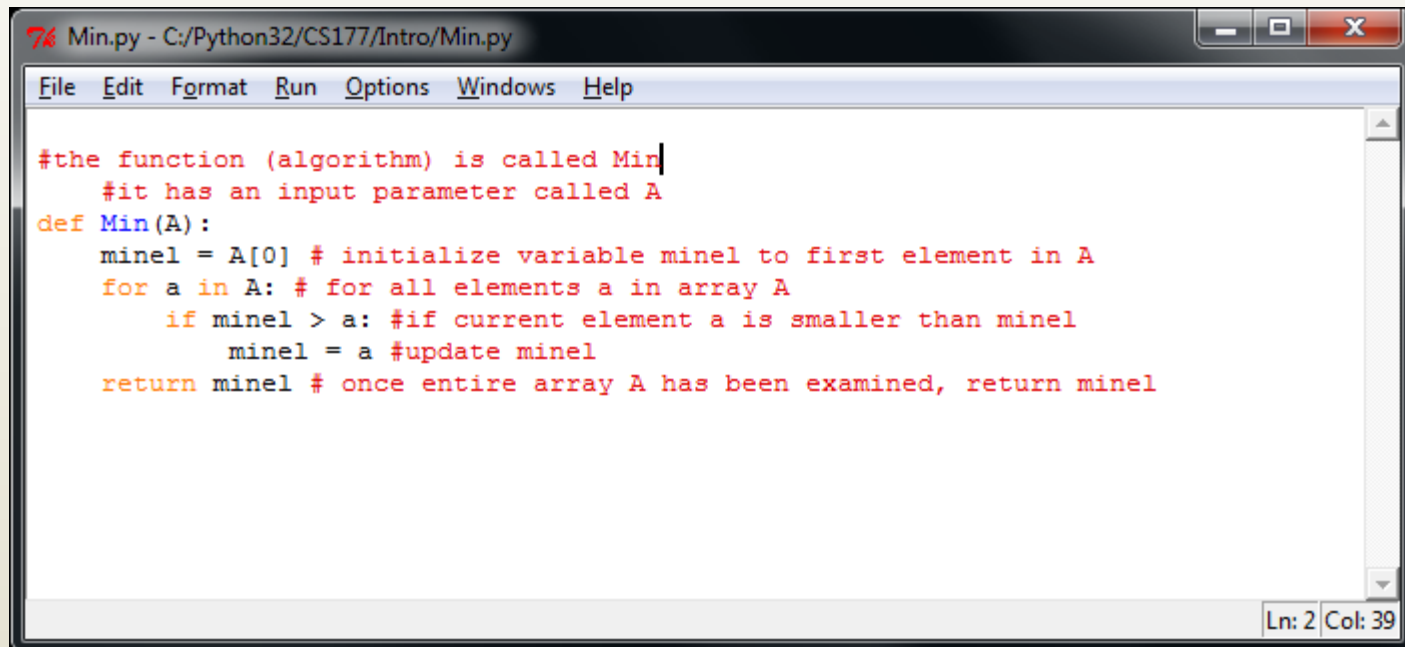


A screenshot of a Python IDE window titled "Min.py - C:/Python32/CS177/Intro/Min.py". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains the following Python code:

```
def Min(A):  
    minel = A[0]  
    for a in A:  
        if minel > a:  
            minel = a  
    return minel
```

The status bar at the bottom right indicates "Ln: 8 Col: 0".

Example 2: minimum in a 1-D array



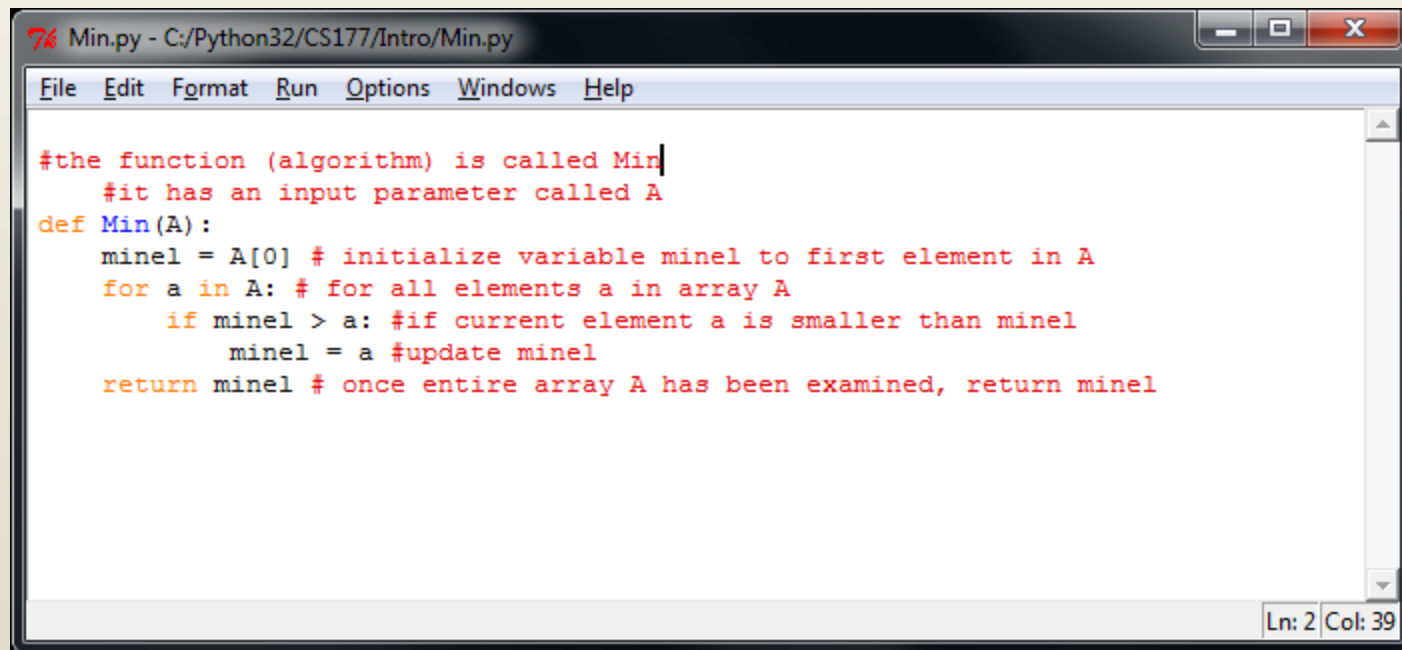
The image shows a screenshot of a Python IDE window titled "Min.py - C:/Python32/CS177/Intro/Min.py". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains the following Python code:

```
#the function (algorithm) is called Min|
#it has an input parameter called A
def Min(A):
    minel = A[0] # initialize variable minel to first element in A
    for a in A: # for all elements a in array A
        if minel > a: #if current element a is smaller than minel
            minel = a #update minel
    return minel # once entire array A has been examined, return minel
```

The status bar at the bottom right indicates "Ln: 2 Col: 39".

Example 2: minimum in a 1-D array

- For loop iterates directly over elements and not over indices



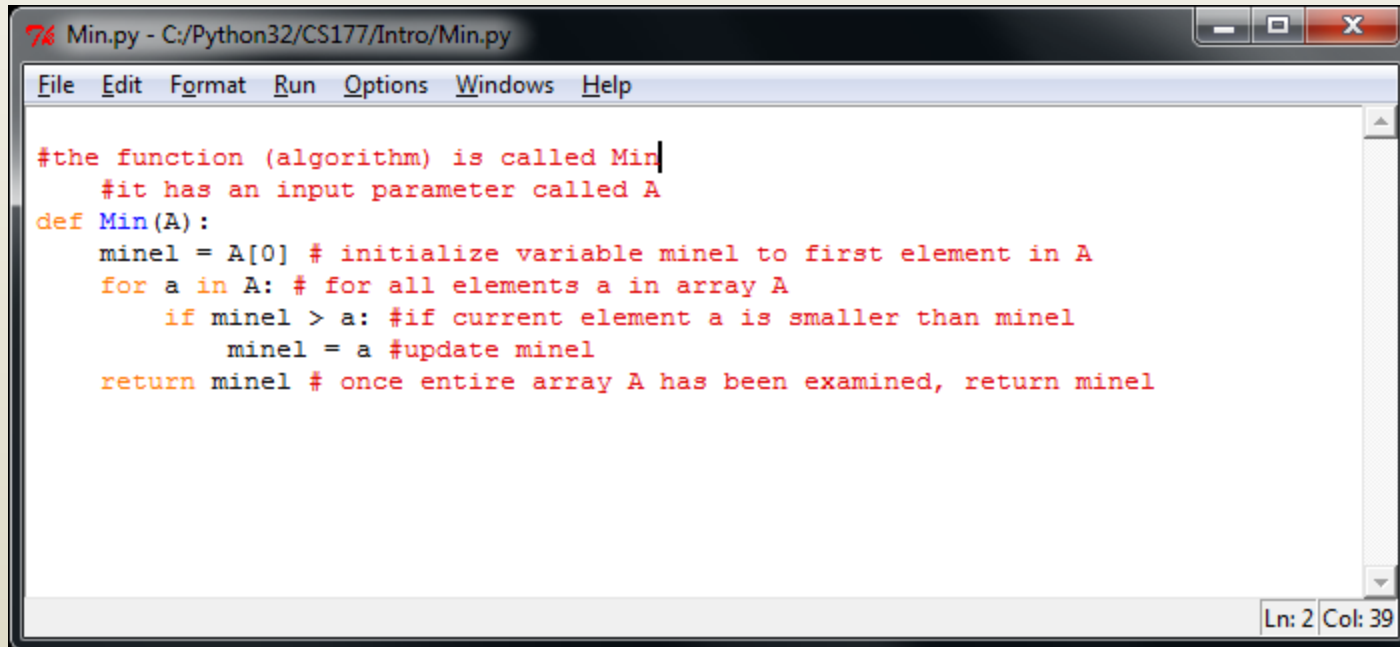
The screenshot shows a Python IDE window titled "Min.py - C:/Python32/CS177/Intro/Min.py". The window contains a Python function named "Min" that takes an array "A" as input and returns the minimum element. The function uses a for loop to iterate directly over the elements of the array "A".

```
#the function (algorithm) is called Min
#it has an input parameter called A
def Min(A):
    minel = A[0] # initialize variable minel to first element in A
    for a in A: # for all elements a in array A
        if minel > a: #if current element a is smaller than minel
            minel = a #update minel
    return minel # once entire array A has been examined, return minel
```

The status bar at the bottom right of the window indicates "Ln: 2 Col: 39".

Example 2: minimum in a 1-D array

- For loop goes over all elements of A, no need to know how many there are
 - Notice that we do not pass in n as a parameter



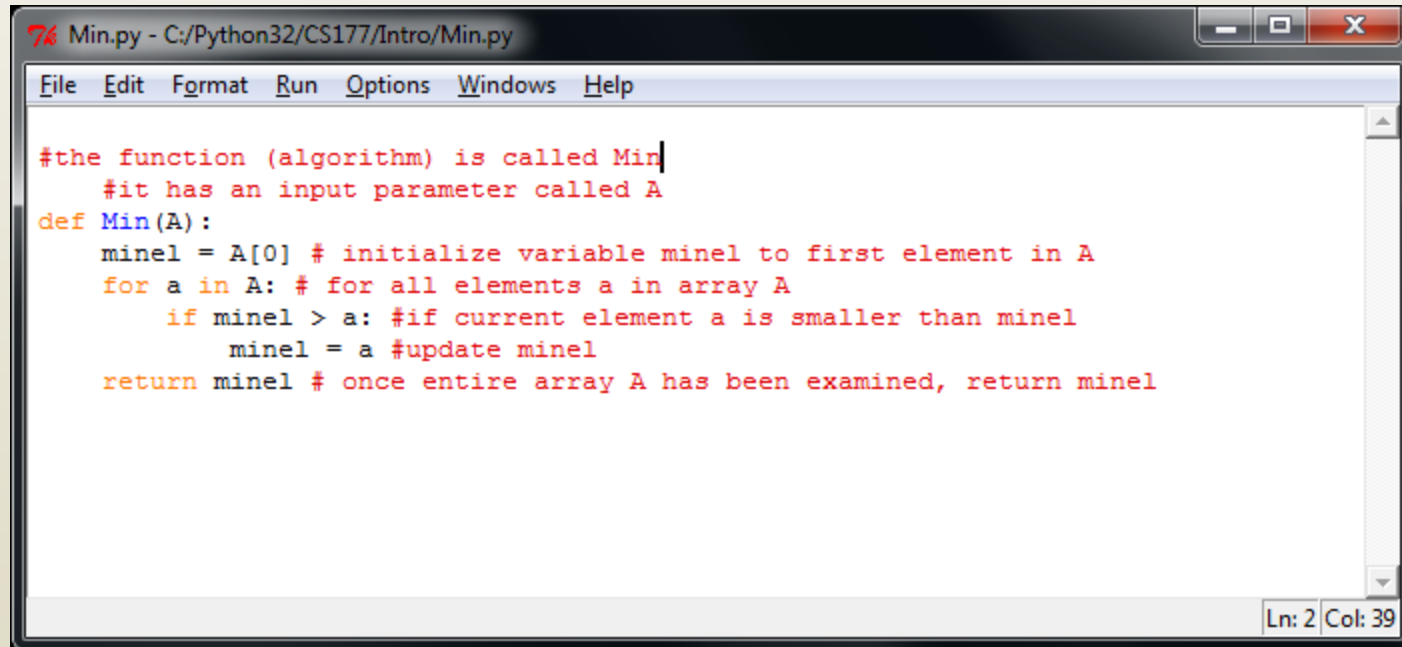
```
Min.py - C:/Python32/CS177/Intro/Min.py
File Edit Format Run Options Windows Help

#the function (algorithm) is called Min
#it has an input parameter called A
def Min(A):
    minel = A[0] # initialize variable minel to first element in A
    for a in A: # for all elements a in array A
        if minel > a: #if current element a is smaller than minel
            minel = a #update minel
    return minel # once entire array A has been examined, return minel

Ln: 2 Col: 39
```

Example 2: minimum in a 1-D array

- Indentation is essential
 - if is part of for loop body, return isn't



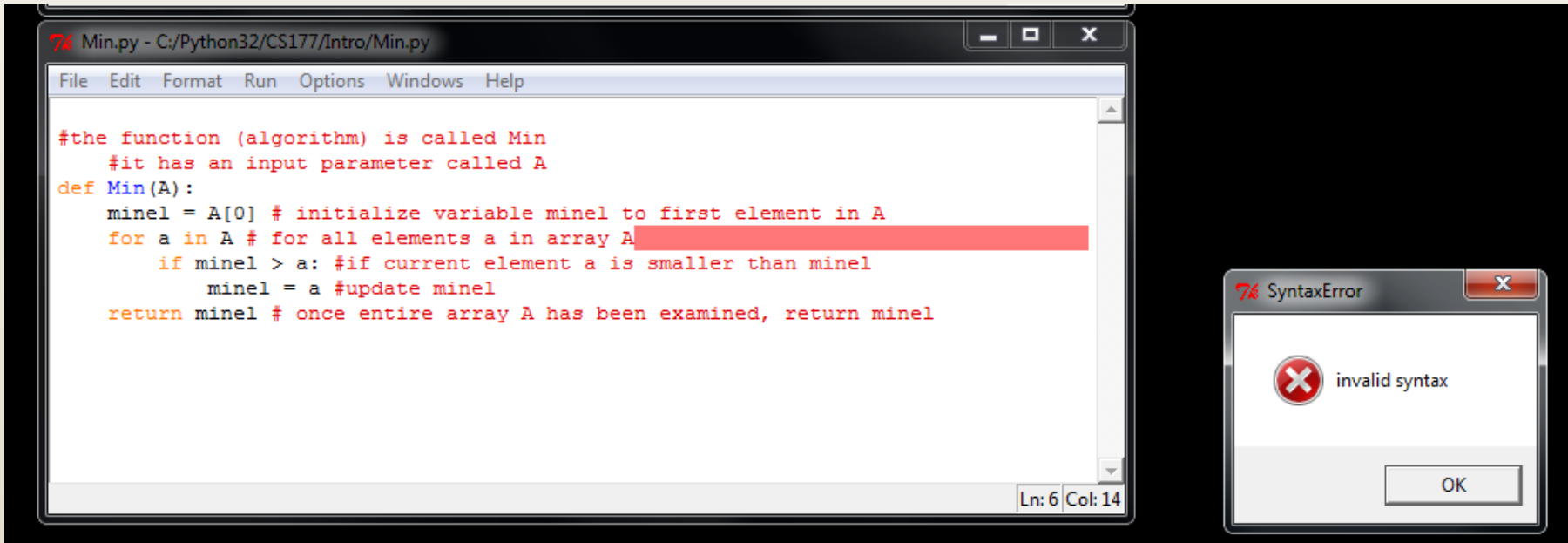
The screenshot shows a Python IDE window titled "Min.py - C:/Python32/CS177/Intro/Min.py". The window contains the following Python code:

```
#the function (algorithm) is called Min
#it has an input parameter called A
def Min(A):
    minel = A[0] # initialize variable minel to first element in A
    for a in A: # for all elements a in array A
        if minel > a: #if current element a is smaller than minel
            minel = a #update minel
    return minel # once entire array A has been examined, return minel
```

The code is color-coded: comments are red, the function definition is blue, and the loop body is orange. The status bar at the bottom right indicates "Ln: 2 Col: 39".

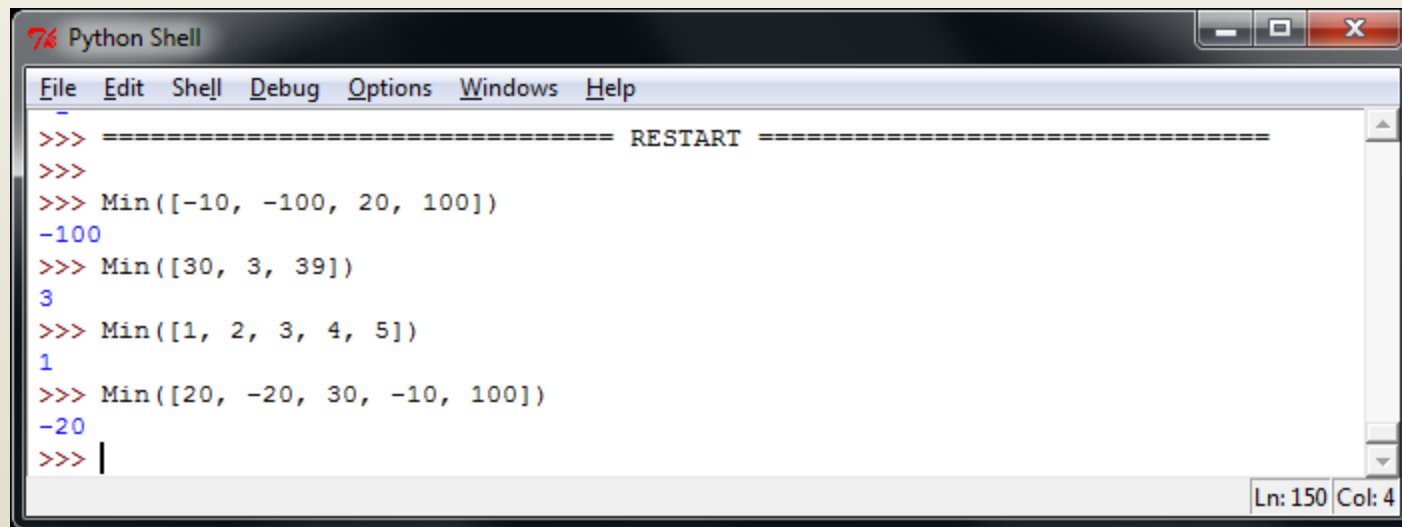
Example 2: minimum in a 1-D array

- Some syntax errors are discovered as module is “run”, but before it is executed
 - Here missing “:” after “for a in A”



Example 2: minimum in a 1-D array

- Then we can call Min from shell interpreter with various parameters A



```
Python Shell
File Edit Shell Debug Options Windows Help
>>> ===== RESTART =====
>>>
>>> Min([-10, -100, 20, 100])
-100
>>> Min([30, 3, 39])
3
>>> Min([1, 2, 3, 4, 5])
1
>>> Min([20, -20, 30, -10, 100])
-20
>>> |
```

Ln: 150 Col: 4