ISE 314X
Computer Programing for Engineers

Chapter 8
Loop Structures and Booleans

Yong Wang
Assistant Professor
Systems Science & Industrial Engineering
Binghamton University

Objectives

- To understand how to implement loops using `for` and `while` statements
- To understand Boolean expressions

BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

For Loops: A Quick Review

```
for <var> in <sequence>:
    <body>
```

- The for loop is a definite loop, meaning that the number of iterations is determined before the loop starts

BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

For Loops: A Quick Review

- Write a program that computes the average of a series of numbers entered by the user
- We don't need to keep track of each number entered
- We only need know the running sum and how many numbers have been added

BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

For Loops: A Quick Review

```
# average1.py

def main():
    n = eval(input("How many numbers do you have?"))
    sum = 0.0
    for i in range(n):
        x = eval(input("Enter a number >>"))
        sum = sum + x
    print("The average of the numbers is", sum/n)

main()
```

For Loops: A Quick Review

```
How many numbers do you have? 5
Enter a number: 32
Enter a number: 45
Enter a number: 34
Enter a number: 76
Enter a number: 45
The average of the numbers is 46.4
```

Indefinite Loops

- Sometimes, we do not know how many iterations we need until all the numbers have been entered
- The *indefinite* loop keeps iterating until certain conditions are met

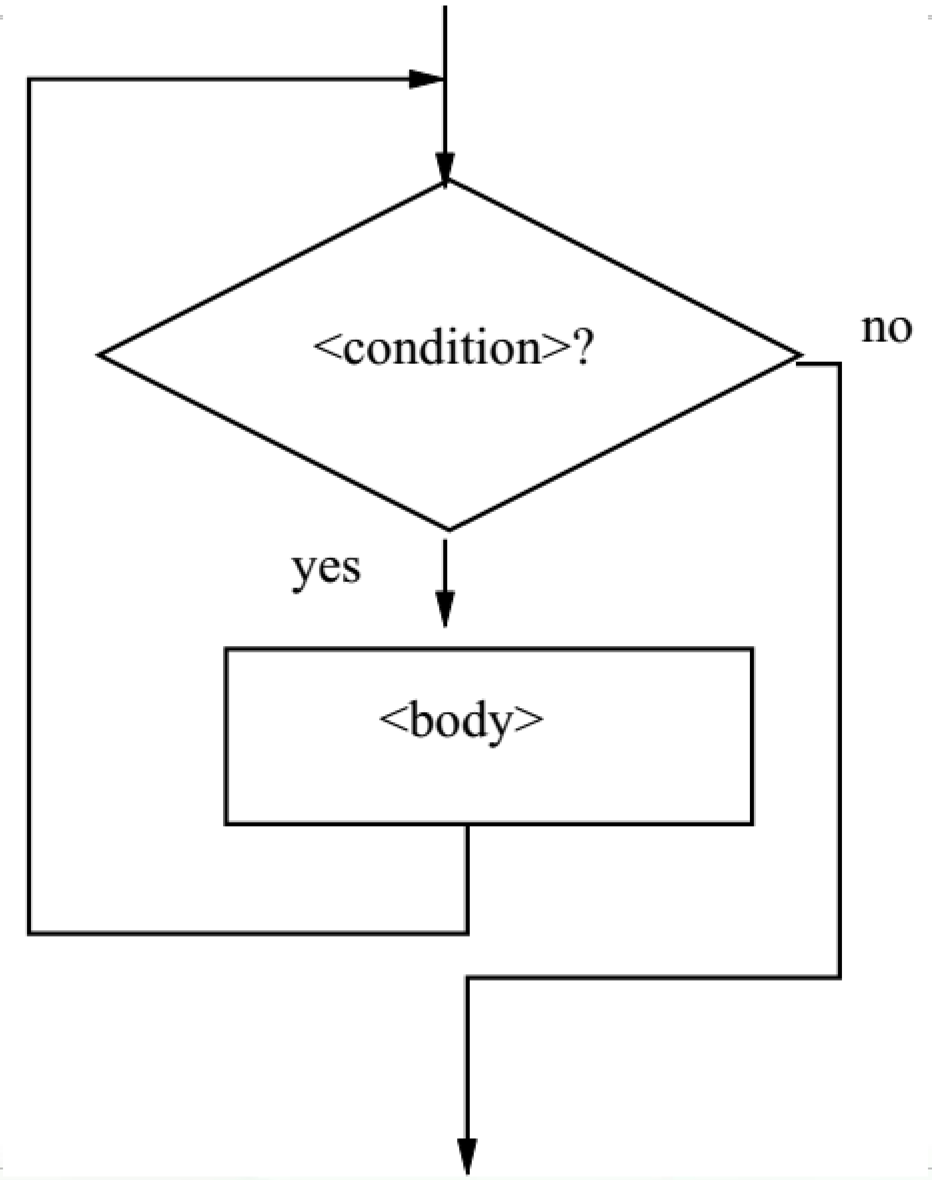
Indefinite Loops

```
while <condition>:
    <body>
```

- `condition` is a Boolean expression, just like in `if` statements
- The body of the loop executes repeatedly as long as the condition remains True
- When the condition is False, the loop terminates

BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

Indefinite Loops



Indefinite Loop

- Count from 0 to 10

```
>>> for i in range(11):
...     print(i)
```

- Which is equivalent to

```
>>> i = 0
>>> while i <= 10:
...     print(i)
...     i = i + 1
```

Indefinite Loop

- Careless use of `while` could cause you trouble

```
>>> i = 0
>>> while i <= 10:
...     print(i)
```

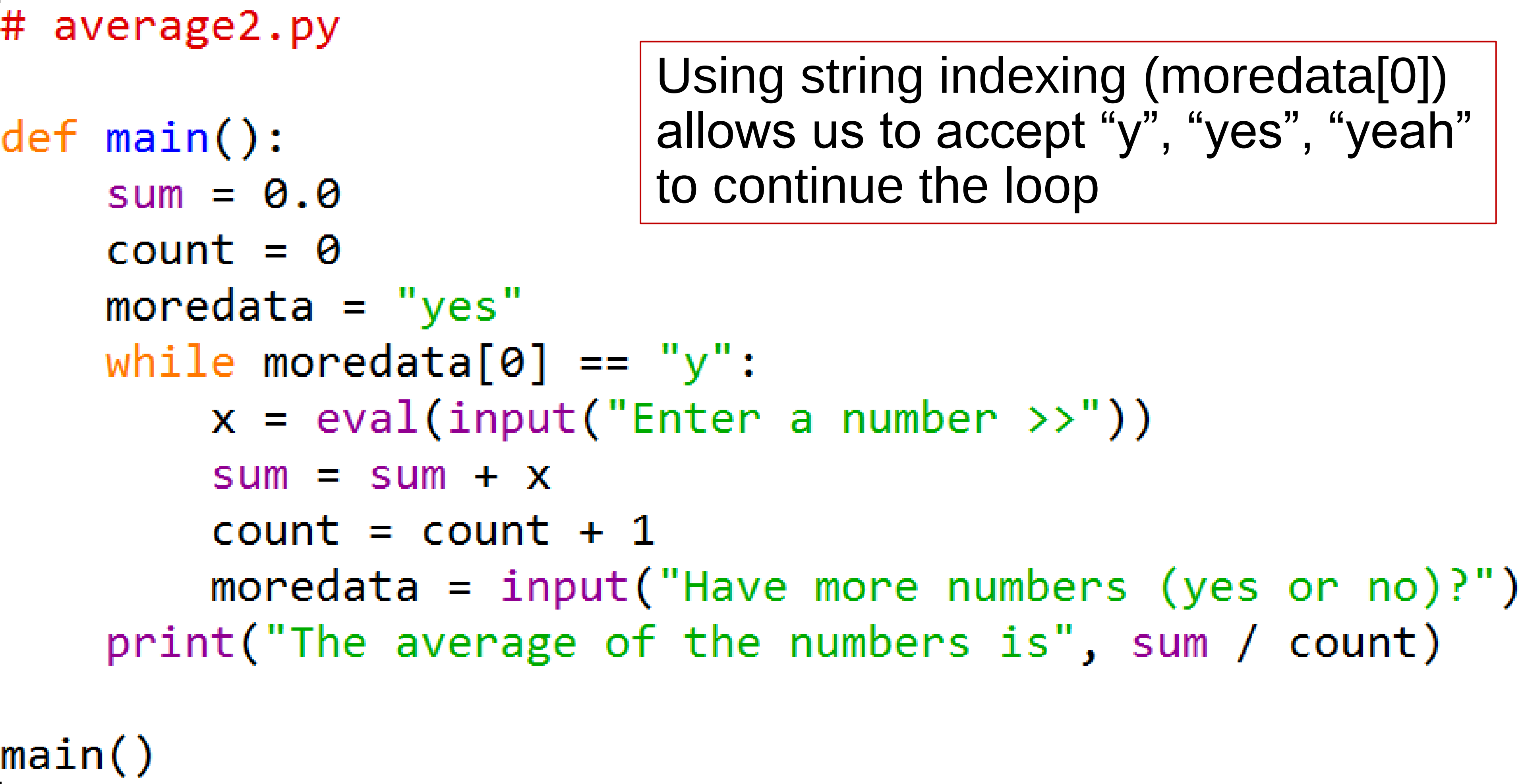
Indefinite Loop

- If you're caught in an infinite loop
  - First, try pressing CTRL+C
  - If that doesn't work, try CTRL+ALT+DEL
  - If that doesn't work, push the reset button

Interactive Loops

- `While` is suitable for *interactive loops*
- At each iteration of the loop, ask the user if there is more data to process

Interactive Loops



```
# average2.py

def main():
    sum = 0.0
    count = 0
    moredata = "yes"
    while moredata[0] == "y":
        x = eval(input("Enter a number >>"))
        sum = sum + x
        count = count + 1
        moredata = input("Have more numbers (yes or no)?")
    print("The average of the numbers is", sum / count)

main()
```

Using string indexing (moredata[0]) allows us to accept “y”, “yes”, “yeah” to continue the loop

Interactive Loops

```
Enter a number: 32
Have more numbers (yes or no)? y
Enter a number: 45
Have more numbers (yes or no)? yes
Enter a number: 34
Have more numbers (yes or no)? yup
Enter a number: 76
Have more numbers (yes or no)? yeah
Enter a number: 45
Have more numbers (yes or no)? nah
The average of the numbers is 46.4
```

BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

Sentinel Loops

- A *sentinel loop* continues to process data until it meets a special value that signals the end
- This special value is called the *sentinel*

BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

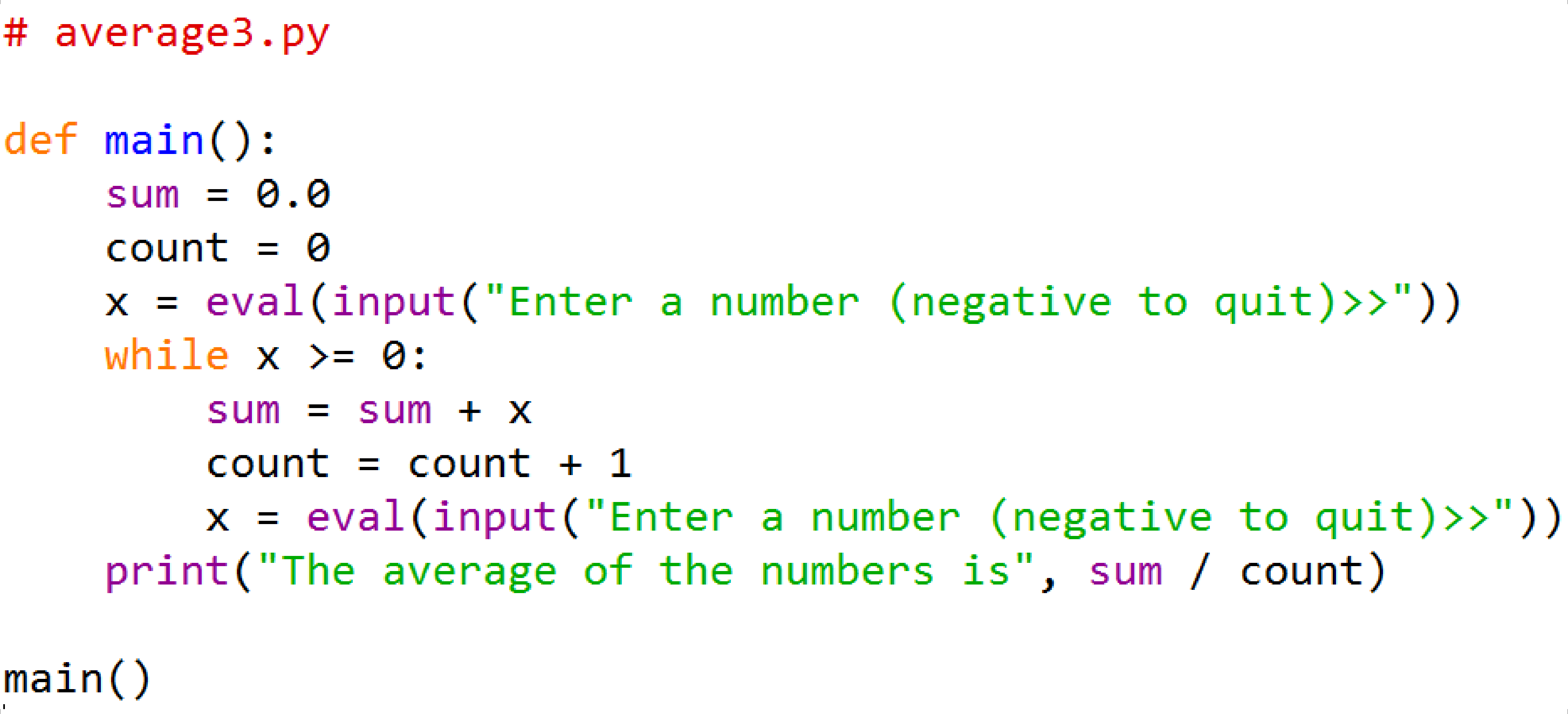
Sentinel Loops

```
get the first data item
while item is not the sentinel
    process the item
    get the next data item
```

Sentinel Loops

- Assume we are averaging test scores
- No score will be below 0
- So a negative number will be the sentinel

Sentinel Loops



```
# average3.py

def main():
    sum = 0.0
    count = 0
    x = eval(input("Enter a number (negative to quit)>>"))
    while x >= 0:
        sum = sum + x
        count = count + 1
        x = eval(input("Enter a number (negative to quit)>>"))
    print("The average of the numbers is", sum / count)

main()
```

Sentinel Loops

```
Enter a number (negative to quit)>> 32
Enter a number (negative to quit)>> 45
Enter a number (negative to quit)>> 34
Enter a number (negative to quit)>> 76
Enter a number (negative to quit)>> 45
Enter a number (negative to quit)>> -1
The average of the numbers is 46.4
```