Chapter 7 Reading Questions

David Letscher

Saint Louis University

CSCI 1300: Introduction to Object-Oriented Programming

Major Questions

- How do loops variables work?
- Types of for loops
- Accumulators
- Nested loops

Loops and Loop Variables

The following are equivalent

```
names = ['Abe','Beth','Carol']
for friend in names:
    print(friend)
```

Loops and Loop Variables

The following are equivalent

```
names = ['Abe','Beth','Carol']
for friend in names:
    print(friend)

names = ['Abe','Beth','Carol']
friend = 'Abe'
print(friend)
friend = 'Beth'
print(friend)
friend = 'Carol'
print(friend)
```

friend is the loop variable. It is defined by the for statement and redefined every time through the loop.

Using Loops to Simplify Code

The following are equivalent

```
print('The price is $10.02')
print('The price is $89.14')
print('The price is $21.53')
print('The price is $45.09')
print('The price is $68.95')
```

Using Loops to Simplify Code

The following are equivalent

```
print('The price is $10.02')
print('The price is $89.14')
print('The price is $21.53')
print('The price is $45.09')
print('The price is $68.95')

prices = [10.02, 89.14, 21.53, 45.09, 68.95]
for amount in prices:
    print('The price is $' + str(amount))
```

When you see identical code repeated or similar code repeated consider using a for loop to simplify.

Index Based Loops

The following are equivalent

```
names = ['Abe','Beth','Carol']
for index, friend in enumerate(names):
    print(index+1, friend)
```

Index Based Loops

The following are equivalent

```
names = ['Abe','Beth','Carol']
for index, friend in enumerate(names):
    print(index+1, friend)

names = ['Abe','Beth','Carol']
index = 0
friend = 'Abe'
print(index+1, friend)
index = 1
friend = 'Beth'
print(index+1, friend)
index = 2
friend = 'Carol'
print(index+1, friend)
```

Range Based Loops

for number in range(10): number takes the values 0, 1, ..., 9 but not 10

It always stops before the upper end of the range is reached

Range Based Loops

for number in range(10): number takes the values 0, 1, ..., 9 but not 10

It always stops before the upper end of the range is reached

for number in range(2,7): number takes the valves 2, 3, 4, 5, 6

With two numbers the first is the starting point

Range Based Loops

for number in range(10):	number takes the values 0, 1, ..., 9 but not 10
It always stops before the upper end of the range is reached	
for number in range(2,7):	number takes the valves 2, 3, 4, 5, 6
With two numbers the first is the starting point	
for number in range(4,10,2):	number takes the values 4, 6, 8
The third number is the step size	

Range Based Loops

for number in range(10):	number takes the values 0, 1, ..., 9 but not 10
It always stops before the upper end of the range is reached	
for number in range(2,7):	number takes the valves 2, 3, 4, 5, 6
With two numbers the first is the starting point	
for number in range(4,10,2):	number takes the values 4, 6, 8
The third number is the step size	
for number in range(5,−1,−1):	number takes the values 5, 4, 3, 2, 1, 0
Be carefull about the stoping point with negative step sizes.	

Reading Quiz

Most difficult questions

Which of the following does the same thing as the following program:

```
s = ''
for item in names:
    s = s + item
```

Reading Quiz

Most difficult questions

Which of the following does the same thing as the following program:

```
s = ''
for item in names:
    s = s + item
```

59% `s = ''.join(names)`

27% Exits with an error

9% `s = ' '.join(names)`

Reading Quiz

Most difficult questions

Which of the following does the same thing as the following program:

```
s = ''
for item in names:
    s = s + item
```

59% `s = ''.join(names)` **Correct answer**

27% Exits with an error

9% `s = ' '.join(names)`

Let's look at this example carefully.

Unwrapping the Loops

```
names = ['Mary', 'Beth', 'Anne']
s = ''
for item in names:
    s = s + item
```

Unwrapping the Loops

```
names = ['Mary', 'Beth', 'Anne']
s = ''
for item in names:
    s = s + item
```

```
names = ['Mary', 'Beth', 'Anne']
s = ''
```

s is now ''

Unwrapping the Loops

```
names = ['Mary', 'Beth', 'Anne']
s = ''
for item in names:
    s = s + item
```

```
names = ['Mary', 'Beth', 'Anne']
s = ''
item = 'Mary'
s = s + item
```

s is now 'Mary'

Unwrapping the Loops

```
names = ['Mary', 'Beth', 'Anne']
s = ''
for item in names:
    s = s + item
```

```
names = ['Mary', 'Beth', 'Anne']
s = ''
item = 'Mary'
s = s + item
item = 'Beth'
s = s + item
```

s is now 'MaryBeth'

Unwrapping the Loops

```
names = ['Mary', 'Beth', 'Anne']
s = ''
for item in names:
    s = s + item
```

```
names = ['Mary', 'Beth', 'Anne']
s = ''
item = 'Mary'
s = s + item
item = 'Beth'
s = s + item
item = 'Anne'
s = s + item
```

s is now 'MaryBethAnne'

Another Accumulator Challenge

What does this print?

```
numbers = [5, 3, 7]
x = 1
for i in numbers:
    x = x * i
print(x)
```

Another Accumulator Challenge

What does this print?

```
numbers = [5, 3, 7]
x = 1
for i in numbers:
    x = x * i
print(x)
```

```
numbers = [5, 3, 7]
x = 1
```

x is now 1

Another Accumulator Challenge

What does this print?

```
numbers = [5, 3, 7]
x = 1
for i in numbers:
    x = x * i
print(x)
```

```
numbers = [5, 3, 7]
x = 1
i = 5
x = x * i
```

x is now 5

Another Accumulator Challenge

What does this print?

```
numbers = [5, 3, 7]
x = 1
for i in numbers:
    x = x * i
print(x)
```

```
numbers = [5, 3, 7]
x = 1
i = 5
x = x * i
i = 3
x = x * i
```

x is now 15

Another Accumulator Challenge

What does this print?

```
numbers = [5, 3, 7]
x = 1
for i in numbers:
    x = x * i
print(x)
```

```
numbers = [5, 3, 7]
x = 1
i = 5
x = x * i
i = 3
x = x * i
i = 7
x = x * i
```

x is now 105

Write your own

Challenge

Write code that will add up all of the values in the list numbers.

Write your own

Challenge

Write code that will add up all of the values in the list numbers.

Solution

```
sum = 0
for i in numbers:
    sum = sum + i
```

Write your own

Challenge

Write code that will add up all of the values in the list numbers.

Solution

```
sum = 0
for i in numbers:
    sum = sum + i
```

Alternate solution

```
sum = 0
for i in numbers:
    sum += i
```

Nested Loops

```
for number in range(1,5):
    for repeat in range(number):
        print('Mom! ', end='')
    print()
```

Nested Loops

```
for number in range(1,5):
    for repeat in range(number):
        print('Mom! ', end='')
    print()
```

See the code in action: `https://goo.gl/ryvSTC`