COMP 141

Nested Loops

Announcements

• Reminder
  – Program 5 has been assigned – due 10/19

Practice From Last Time

Write a program that starts off asking the user how much money they have in their bank account. Next, add a menu to let the user add money, subtract money, or quit the ATM program. Let the user keep using the ATM as long as they want (until they choose to quit). Prevent the user from withdrawing more money than they have in their account. Use input validation to prevent the user from typing in a negative amount of money.

Nested Loops

• Write a loop to print out “*****”
  for j in range(5):
    print('', end='')
    print()

• Write code to print out the following.
  ***** for i in range(4):
    ***** for j in range(5):
      ***** print('', end='')
      ***** print()
Nested Loops Examples

- Input from file
  - Keep inputting next line and take average of numbers on the line until the line == ‘’
- Want to create a ‘clock’ object that needs to go through every second of every minute of every day
- Iterate through every box in a 2-D grid

```
input_file = open('input.txt', 'r')
for line in input_file:
    numbers = line.split()
    average = sum(map(int, numbers)) / len(numbers)
    print(average)
```

Nested Loops

- Definition - A loop that is inside another loop.
- An inner loop goes through all of its iterations for every single iteration of an outer loop.
- Inner loops complete their iterations faster than outer loops.
- To get the total number of iterations of a nested loop, multiply the number of iterations of all the loops

```
for i in range(4):
    for j in range(5):
        print('***', end='***')
        print('*****', end='*****')
```

Syntax

```
for iterating_var in sequence:
    statement(s)
```

```
while expression:
    while expression:
        statement(s)
        statement(s)
```

```
while expression:
    for iterating_var in sequence:
        statement(s)
        statement(s)
```

```
while expression:
    for iterating_var in sequence:
        statement(s)
```

```
while expression:
    for iterating_var in sequence:
        statement(s)
```

```
Example 1 – Times Tables

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td>21</td>
<td>24</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>16</td>
<td>20</td>
<td>24</td>
<td>28</td>
<td>32</td>
<td>36</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>12</td>
<td>18</td>
<td>24</td>
<td>30</td>
<td>36</td>
<td>42</td>
<td>48</td>
<td>54</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>14</td>
<td>21</td>
<td>28</td>
<td>35</td>
<td>42</td>
<td>49</td>
<td>56</td>
<td>63</td>
<td>70</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>32</td>
<td>40</td>
<td>48</td>
<td>56</td>
<td>64</td>
<td>72</td>
<td>80</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>18</td>
<td>27</td>
<td>36</td>
<td>45</td>
<td>54</td>
<td>63</td>
<td>72</td>
<td>81</td>
<td>90</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
</tr>
</tbody>
</table>

for i in range(1, 11):
    for j in range(1, 11):
        print((i*j), end=' ')
print()

Example 2 – Dependent Loops

for i in range(1, 8, 3):
    for j in range(8, i, -2):
        print(j, end='')

8 6 4 2 8 6 8

Example 3 – Prime Numbers

```python
i = 2
while i < 100:
    prime = True
    for j in range(2, i):
        if i % j == 0:
            prime = False
            break
    if prime:
        print(i, "is prime")
    i = i + 1
```

```
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
13 is prime
17 is prime
19 is prime
23 is prime
29 is prime
31 is prime
37 is prime
41 is prime
43 is prime
47 is prime
53 is prime
59 is prime
61 is prime
67 is prime
71 is prime
73 is prime
79 is prime
83 is prime
89 is prime
97 is prime
```

Example 4 – Clock Example

Print every second of every minute of every hour in 1 day.

```python
for hours in range(0,24):
    for minutes in range(0,60):
        for seconds in range(0,60):
            print(hours, ":", minutes, ":", seconds)
```

```
0 : 0 : 0
0 : 0 : 1
0 : 0 : 2
0 : 0 : 3
........
23 : 59 : 57
23 : 59 : 58
23 : 59 : 59
```

for hours in range(0,24):
    for minutes in range(0,60):
        for seconds in range(0,60):
            print(hours, ":", minutes, ":", seconds)
Writing Nested Loops

Designing nested loops

- Design the outer loop without worrying about what goes inside
- Design what goes inside, ignoring the outer loop.
- Put the pieces together, preserving the nesting.

Turtle Functions

http://docs.python.org/2/library/turtle.html

- forward(pixels), back(pixels)
- right(degrees), left(degrees)
- penup(), pendown()
- goto(x, y) - moves the turtle to a new location on the grid
- pencolor(color), bgcolor(color)
  - color = 'red', 'blue' or (r,g,b) where r,g,b are values between 0-1
- shape('shape') - triangle, circle, etc.
- stamp() - Stamp a copy of the turtle shape onto the canvas at the current turtle position
- fillcolor(color) - use with shapes

Designs by Turtle

```
bgcolor("black")
pencolor("red")
penize(3)
ne = 36
size = 20
for i in range(ne):
  right(360./ne)
  for j in range(ne):
    right(360./ne)
  forward(size)
```
Designs by Turtle

```python
bcolor("black")
shape("triangle")
phi = 0.89637
for i in range(5):
s = 20
c = 1
for i in range(20):
    shapesize(s)
    fillcolor(c, 0.5, 1-c)
    stamp()
s *= r
c *= r
right(phi)
right(360/5)
```