## Prior Chapter Review Questions

1. Write a **for** loop that will print out a horizontal line of ten asterisks (\*).
2. Write two nested **for** loops that will print a 10x10 box of asterisks.
3. Write Python code that will **create** an array of 100 zeros.
4. What is the difference between a class and an object?
5. What is the difference between a function and a method?
6. Write a function that prints your favorite number.
7. Call the function that prints your favorite number.
8. Write a function that takes three numbers and returns the average.
9. Programming classes:
   1. Write code for a class called Ball. Give it attributes for its position, and its velocity.
   2. Create a method called update() that will move the ball’s position according to its velocity.
   3. Create an instance of Ball, set its attributes.
   4. Create a “for” loop that will call the update() method on ball 10 times, and print the ball’s position.

# Sorting Chapter Review

1. Write code to swap the values 25 and 40.

|  |
| --- |
| list = [55, 41, 52, 68, 45, 27, 40, 25, 37, 26] |
|  |

1. Write code to swap the values 2 and 27.

|  |
| --- |
| list = [27, 32, 18, 2, 11, 57, 14, 38, 19, 91] |
|  |

1. Why does the following code not work?

|  |
| --- |
| list = [70, 32, 98, 88, 92, 36, 81, 83, 87, 66]  temp = list[0]  list[1] = list[0]  list[0] = temp |

1. Show how to perform a selection sort on the following numbers:

97 74 8 98 47 62 12 11 0 60

1. Take the following code and fill in the blanks:

# The selection sort

def **selection\_sort**(list):

# Loop through the entire array

for curPos in range( ):

# Find the position that has the smallest number

# Start with the current position

minPos =

# Scan right

for scanPos in range(curPos+1, ):

# Is this position smallest?

if list[scanPos] < list[minPos]:

# It is, mark this position as the smallest

minPos = scanPos

# Swap the two values

1. Show how to perform a insertion sort on the following numbers:

97 74 8 98 47 62 12 11 0 60

1. Take the following code and fill in the blanks:

def **insertion\_sort**(list):

# Start at the second element (pos 1).

# Use this element to insert into the

# list.

for keyPos in range(1, len(list)):

# Get the value of the element to insert

keyValue =

# Scan to the left

scanPos = keyPos – 1

# Loop each element, moving them up until

# we reach the position

while (scanPos >=0) and ( ):

list[scanPos+1] = list[scanPos]

# Everything's been moved out of the way, insert

# the key into the correct location

list[scanPos+1] =

1. Explain what **minPos** does in the selection sort.
2. Explain what **curPos** does in the selection sort.
3. Explain what **scanPos** does in the selection sort.

1. Explain what **keyPos** and **keyValue** are in the insertion sort.
2. Explain **scanPos** in the insertion sort.
3. Modify the sorts to print the number of times the inside loop is run, and the number of times the outside loop is run.