Prior Chapter Review Questions

1. Write a **for** loop that will print out a horizontal line of ten asterisks (*).

```
for i in range(10):
    print("*", end="")
```

2. Write two nested **for** loops that will print a 10x10 box of asterisks.

```
for row in range(10):
    for i in range(10):
        print("*", end="")
    print()
```

3. Write Python code that will **create** an array of 100 zeros.

```
my_array=[]

for i in range(100):
    my_array.append(0)
```

4. What is the difference between a class and an object?

Object is an instance of a class.

For example, “Bob” is an instance of a human

5. What is the difference between a function and a method?

Method is simply a function that is inside of a class.

6. Write a function that prints your favorite number.

```
def my_favorite_number():
    print("7")
```

7. Call the function that prints your favorite number.

    ```
    my_favorite_number()
    ```

8. Write a function that takes three numbers and returns the average.

```
def avg(a,b,c):
    return (a+b+c)/3
```

9. Programming classes:
   a. Write code for a class called Ball. Give it attributes for its position, and its velocity.
   b. Create a method called update() that will move the ball's position according to its velocity.
   c. Create an instance of Ball, set its attributes.
   d. Create a "for" loop that will call the update() method on ball 10 times, and print the ball's position.

```
class Ball():
    x=0
    y=0
    change_x=0
    change_y=0

    def update(self):
        self.x += self.change_x
        self.y += self.change_y

my_ball=Ball()
my_ball.x=4
my_ball.y=6
my_ball.change_x=3
my_ball.change_y=4

for i in range(10):
    my_ball.update()
    print(my_ball.x,my_ball.y)
```

Sorting Chapter Review

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

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

```
list = [55, 41, 52, 68, 45, 27, 40, 25, 37, 26]
print(list)
temp = list[7]
list[7]=list[6]
list[6]=temp
print(list)
```

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

```
list = [27, 32, 18, 2, 11, 57, 14, 38, 19, 91]
print(list)
temp = list[0]
list[0]=list[3]
list[3]=temp
print(list)
```

12. 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
```

Swap should look like:

```
temp = list[0]
list[0] = list[1]
list[1] = temp
```

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

97	74	8	98	47	62	12	11	0	60
0	74	8	98	47	62	12	11	97	60
0	8	74	98	47	62	12	11	97	60
0	8	11	98	47	62	12	74	97	60
0	8	11	12	47	62	98	74	97	60
0	8	11	12	47	62	98	74	97	60
0	8	11	12	47	60	98	74	97	62
0	8	11	12	47	60	62	74	97	98

14. 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(len(list)):

        # Find the position that has the smallest number
        # Start with the current position

        minPos = curPos

        # Scan right
        for scanPos in range(curPos+1,  len(list)      ):

            # Is this position smallest?
            if list[scanPos] < list[minPos]:

                # It is, mark this position as the smallest
                minPos = scanPos

        # Swap the two values
        temp = list[minPos]
        list[minPos]=list[curPos]
        list[curPos]=temp
```

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

97	74	8	98	47	62	12	11	0	60
74	97	8	98	47	62	12	11	0	60
8	74	97	98	47	62	12	11	0	60
8	74	97	98	47	62	12	11	0	60
8	47	74	97	98	62	12	11	0	60
8	62	47	74	97	98	12	11	0	60
8	12	62	47	74	97	98	11	0	60
8	11	12	62	47	74	97	98	0	60
0	8	11	12	62	47	74	97	98	60
0	8	11	12	60	62	47	74	97	98

16. 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 = list[keyPos]

        # Scan to the left
        scanPos = keyPos – 1

        # Loop each element, moving them up until
        # we reach the position

        while (scanPos >=0) and (list[scanPos] > keyValue):
            list[scanPos+1] = list[scanPos]
            scanPos -= 1

        # Everything's been moved out of the way, insert
        # the key into the correct location
        list[scanPos+1] = keyValue
```

17. Explain what `minPos` does in the selection sort.

    Holds the position of the smallest element on each scan

18. Explain what `curPos` does in the selection sort.

    The current position that we are looking to swap the smallest element into.

19. Explain what `scanPos` does in the selection sort.

    The scan position is where we are currently looking to find the smallest element as we scan the array for it.

20. Explain what `keyPos` and `keyValue` are in the insertion sort.

    keyPos is position of the new element we are looking to insert into the list.
    keyValue is the value of that element

21. Explain `scanPos` in the insertion sort.

    Holds where we scan right to left, floating value up until we reach a value that is less than the keyValue.

22. Modify the sorts to print the number of times the inside loop is run, and the number of times the outside loop is run.

Selection Sort:

```
Outside loop: 50
Inside loop: 1225

Insertion Sort
Outside loop: 49
Inside loop: 609
```