

## SPIS Lecture, Week 2, Monday 10:15

### Counting Exercises:

1. Call out increasing numbers starting with 1 to count the number of students in the room
  - a. Counting in a loop
  - b. 48 student took 48 rounds (or 48 loop iterations)
2. More efficient counting:
  - a. Each student stands and has number 1.
  - b. Each round:
    - i. Find a standing student
    - ii. Add numbers
    - iii. Oldest remains standing
  - c. 48 students took 6 rounds (or 6 loop iterations)
3. Counting recursively
  - a. In turn, each student turn their chair to the student next to them, points to them and asks the question, "How many students are you away from the last student?"
    - i. While chair is turned, this simulated that the function is still executing and the student who asked the question is still waiting for an answer.

- ii. All students with chairs turned simulate a “stack” of function calls tracked on the run-time stack.
- b. The last student is 1.
  - i. This is the base case of the recursion
- c. In turn, each student responds to their caller by adding one to the number called out by the prior student and then “return”s their chair to its original position.
  - i. This is simulating the recursive function returning and the calling function resuming executing and returning to its caller

### Observations:

Loop – performing the same operating on a series of elements, as in counting 1.

- No termination means an infinite loop.

Recursion – performing a piece of a task, recursively calling a method to complete the task.

- A base case is needed,
- Perform task once, delete to recursive function to perform remaining task,
- No termination means infinite recursion.

Both recursion and loops give you repeatability.

- Choice is often “loop or recursion” not “loop of recursion.”
- Recursion often involves less code

## Arrays:

- Multiple item allocation of memory, each item is identical in size to the others.
  - o Example: parking lot spaces, chairs in the lecture room
- Each array element has an index to location that item in the array.
- The array’s index starts with 0 since the first item is 0 elements from the beginning.

Python code to print a string:

```
print “abc”
```

-----

Output: abc

Python code to print a string

```
var = "abc"  
index = 0  
while index < len (var):  
    print (var[index], end="")  
    index = index + 1
```

-----

Output: abc

Python code to print a string

```
def print_in_a_loop (var):  
    index = 0  
    while index < len (var):  
        print (var[index], end="")  
        index = index + 1  
print_in_a_loop ("abc\n")
```

-----

Output: abc

Python code to print a string

```
def print_string (var, index):
```

```
    if index < len (var):  
        print (var[index], end="")  
        print_string (var, index + 1) # recursive call
```

```
print_string ("abc\n", 0)
```

-----

Output: abc

Python code to print a string

```
def print_string (var, index):
```

```
    if index < len (var):  
        print_string (var, index + 1) # recursive call  
        print (var[index], end="")
```

```
print_string ("abc\n", 0)
```

-----

Output: (string is printed in reverse order, newline printed first)

cba