Recursions

- Alternative to loops/iteration
- Process where a function calls itself
- Recursive Functions
  - Functions that call themselves
- Two Cases:
  - Base Case
  - General Case

Base Case

- Solves a part of the problem
- No more recursive function calls are made
- Usually, returns a constant
- Every recursive function must have a base case
- If no base case, recursion repeats infinitely, much like an infinite loop
- Equivalent to the condition of a loop

General Case

- Reduces the size of problem for the next recursive call
- Equivalent to the update/increment part of the loop
- General case must reduce the problem until it reaches the base case

Designing Recursive Functions

1. Determine the base case
2. Determine the general case
3. Combine base case & general case to form the recursive function

Limitations of Recursion

- Extensive overhead due to numerous function calls
- Each function call requires the allocation of memory for the new function call
  - Computer may eventually run out of memory