ITP 365

Lecture 9

2/7/17

Even More Recursion

* Can use recursion to draw squares over and over again
* Base case -> width <= 0
* Sum #’s of vector
  + Iteratively
    - Int sum = 0;
    - For (int I = 0; I < numbers.size(); i++)
      * Sum += numbers[i]
    - Return sum;
  + **Recursively**
    - Vector of numbers
    - Summing it all up means adding 1st number + vector of numbers after that
    - Then take 2nd number + vector of numbers after that, etc.
    - Base case?
      * If startIndex == endIndex
    - **Need to add a “helper” function when you need to keep track of indices to do the recursion but you don’t need indices for overall program**
      * **So the recursion actually happens in helpfer function**
    - Int sumOfVectorPart(Vector<int>& numbers, unsigned start, unsigned end)
      * // Base case: if start == end
      * if (start == end){
        + //return number at end index
        + return numbers[end];
      * else{
        + // add number at start index + a new part of the remaining indices
        + return numbers[start] + sumOfVectorPart(numbers, start+1, end)
    - Then add this helper function to original function
      * Int sumOfVector(Vector<int>& numbers)
        + Return SumOfVectorPart(numbers, 0, numbers.size() – 1)