## GWC Level 3, Week 7

Arrays and Recursion

Katie Tooher and Carmen Seda

#### WIT Shout-Out of the Week: Susan Kathy Land

- Ms. Land is the current president of the Institute of Electrical and Electronics Engineering Society
  - A society for advocating the advancement of technology and innovation for the benefit of humanity
- She Graduated with a BS from The University of Georgia and got a MS from Florida Institute of Technology
- She is a Program Manager for the Missile Defense Agency and is responsible for advancing the Command, Control, Battle Management & Communications (C2BMC) Program mission objectives

### **Meet the 2019 IEEE Presidential Candidates**







## Warm-up

Group - Code Last Week's Dice Rolling Problem

#### How to generate a Random Number

# Random random = new Random(); int randomnum = random.Next(0,6);

#### Activities: Dice Rolling Simulation

- Write a program that asks a user for a "yes" or "no" answer to the question: Would you like to roll the dice? In a while loop until a person says "no"
- If a user says yes, then call a method named "roll" that will generate a random number between 0 and 6 and then return that integer **this method should be public**
- If a user says no, then the program quits
- Use the random number generator code from the previous slide to help you with the roll method



#### **Recursion Definition**

- <u>Recursion:</u> a time when a method repeats itself -whenever a method is called within itself
- We use recursion to solve large problems that can be broken up into a series of smaller problems
- In Recursion there are two cases to consider
  - $\circ$   $\,$  The base case and the general or recursive case

## **Consider this Problem**

 We want to solve a factorial problem for any number using a recursive style of programming.

#### **Base Case**

 <u>Base Case</u>: The simplest form the problem can be solved in -- this section does not use recursion and each recursive program must make use of at least one base case

• What might the base case of the Factorial Problem be?

#### General or Recursive Case

- After the base case comes the recursion
- <u>The Recursive Case</u>: the more complex portion of the program that is found to be applicable to all sections of the problem
  - This is the portion which calls it's own method to complete the task by breaking the problem into smaller subproblems or base cases
- How does this relate to the Factorial Problem?

## **Group Activity!**

#### Using Recursion -- Factorial Problem



#### Arrays

- Array a collection of elements
   Ex: a book shelf holds many books
- To declare an array, you need the data type of the elements being stored, brackets, and an array name
- To initialize an array, you need to use the keyword "new" and the **size of the array**

Declaring an array Initializing an array int[] num; num = new int[10];



- Green boxes represent <u>Elements:</u> or items in the list
- Peach bar on top with numbers represent the *indexes* or *locations* of the elements in that array

### **Declaring or Creating Arrays**

- Each element has an "<u>address" (or index)</u>
   the spot in the array where an <u>element</u> is stored
- <u>Elements</u> are accessed by calling the address
   Call by using the array name and brackets
- The first element of an array is address 0
  - Ex: "hello" what is the address of each letter?
- Strings can be treated like arrays because they are an array of characters **WHAT IS STORED IN C**?
  - string s = "hello";
  - char c = s[0];

#### **Assigning Values to Arrays**

- Can assign / change values of elements by calling them
- Can initialize and assign variables at same time of declaration

 Call the elements
 Declare and create

 int[] num = new int[3];
 int[] num = {2,8,74};

 num[0] = 2;
 Declare, initialize, and create

 num[1] = 8;
 int[] num = new int[] {2,8,74};

 num[2] = 74;
 Declare, initialize, and create

 int[] num = new int[] {2,8,74};

 num[2] = 74;

 Declare, initialize, and create

 int[] num = new int[3] {2,8,74};

## Group Activity!



Fill in the values of an array with a size of 10 using a for loop. Values should be even numbers up to 20 in numerical order.

## Your Turn!

## Activity

Create an array that has a length of 15. Assign the following values to the array: 16, 22, 89, 43, 56, 33, 1, 20, 99, 104, 67, 75, 13, 234, and 47. Find the SUM of all the elements in the array.

Hint: use a for-each loop.

#### **Recursion Activity**

- A person can walk up stairs either 1 or 2 steps at a time
- Count the number of ways a person can reach the top of the stairs for any n number of stairs using only one or two steps at a time

Example:

For 1 step there is 1 way, for 2 steps there are 2 ways, and for 10 steps there are 89 ways.

• Hint: looks very similar to the factorial problem