

## Programming Project - Recursion

- 1) Review the material about recursion
- 2) If you have not done so already, download the source code from <https://github.com/CGCC-CS/205activity4.git>. The file Project4.java is a driver class that tests the methods from the activity & the methods below.
- 3) Add the following methods to your Recursion class:
  - A. A *recursive* Java method called `power(x, y)` to find  $x$  raised to the power of  $y$ . You can assume that  $y \geq 0$ . Note that your method must be recursive. You will receive no credit if you call `Math.pow()`.
  - B. Add a *recursive* Java method called `balance(x, y)` to compute the floor of the average of  $x$  &  $y$  as follows:
    - If the two parameters are within 1 of each other, return the smaller number
    - Otherwise, subtract one from the larger parameter and add one to the smaller parameter and return `balance` of those two numbers
  - C. A *recursive* Java method called `Ackermann(m, n)` to compute the Ackerman function defined as follows:
$$A(m, n) = \begin{cases} n + 1 & \text{if } m = 0 \\ A(m - 1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m - 1, A(m, n - 1)) & \text{if } m > 0 \text{ and } n > 0. \end{cases}$$

The Ackermann function grows very quickly, so you may run into stack overflows when  $m > 3$ . You can see a table of correct values here: [https://en.wikipedia.org/wiki/Ackermann\\_function](https://en.wikipedia.org/wiki/Ackermann_function)
  - D. A method called `playGuessingGame(m)` that lets the user play a guessing game. The method will pick a random number between 0 &  $m$  and prompt the user enter a guess from 0 to  $m$ . If the user does not guess the number then they should be told whether their guess was higher or lower than the number then display a new range to choose from. Your `playGuessingGame` method should call a *recursive* helper method that prints the range the user should guess between.
- 4) Uncomment all the test code in the Project4 driver class and include sample output for all the methods you wrote for the activity & project in your submission document.

### Submission requirements:

- Include your name as a comment at the top of each source code file
- Make good use of whitespace/comments to make your implementation clear.
- In a well-formatted .doc, .pdf, or .txt file, briefly describe your implementation, give sample output, and include all 4 methods from part 3
- Zip your entire Eclipse project (including .class files). Do not use .rar.
- Include your first and last name in the .zip filename
- Upload your implementation/output document & zipped project **separately** to Canvas
- (Optional) Turn in a hard copy of your implementation document

Be prepared to demo your project in class following the due date.