Java Crash Course

1. Make a new project:
    a. File -> New Project
    b. Choose “Java Application”. Click Next.
    c. Choose a project name (doesn’t matter too much) and a project folder (anywhere you want on your computer).
    d. ***important***: Click Create Main Class and call it CrashCourse. This will automatically create a file called CrashCourse.java and give you a public static void main function.

2. In main, write code to have the user type in integers from the keyboard until they enter -1. Compute the running total of all the numbers as they are entered. Display their sum at the end.
    a. Hints: at the top of the program, put import java.util.*;
    b. in main, make a new scanner that reads from System.in.
    c. Use a while loop and scanner.nextInt() inside to read integers.

3. Modify your code so that all the integers are added to an ArrayList<Integer> as they are typed in (except the -1 sentinel).
    Print out all the integers after the -1 is typed. Try this two ways. One with System.out.println(list) and one with the enhanced for loop.

4. Add a function private static long fib(int n) that computes the n’th Fibonacci number. Write this the standard, slow, recursive way:
    fib(0) = 0, fib(1) = 1, fib(n) = fib(n-1)+fib(n-2). Change your input loop so that whatever number *n* the user types in, you print out fib(n). What is the maximum Fibonacci number you can compute before you get an error?

5. Copy fib(n) to a new function called mfib(n). This will be a memorized version of Fibonacci. Add a HashMap<Integer, Integer> as a static field to your class. This will store the cached Fibonacci values.

    Alter your Fibonacci method so it does the following:
    – For mfib(n):
    – if n = 0 or n = 1, return n
    – Check if n is a key in the hashtable.
        If it is, get the corresponding value and return it.
        If it's not, then
        – compute v = mfib(n-1) + mfib(n-2)
        – put the mapping from n to v in the hashtable
        – return v

    Change your main function to call mfib rather than regular fib. What is the highest fib number you can compute now?

6. Suppose we want to change our main function so that we print an error message if the user enters the same number twice (not necessarily back to back, just any time). We will do this using a HashSet to store all the numbers previously entered.

    In main(), add a HashSet variable that will store integers. As each number *n* is entered, check to see if it’s in the set. If it is, print an error message. If it isn’t, add it to the set and print mfib(n) as normal.