

1. Write a program that uses two **recursive** methods to do the following:
  - a. The first method calculates and returns the value of  $2^n$  for  $n \geq 1$ . For example, if the user enters 5, then  $2^5 = 32$ .
  - b. The second method calculates and returns the sum of all the positive integers from 1 up to the number passed as the argument. For example, if 5 is passed as the argument, the method will return the sum of 1, 2, 3, 4, and 5.
  - c. In the main method, call these methods with user input.
2. Assume that there are  $N$  programming languages to learn, and you must learn  $x$  of them. How many possible combinations you can learn?

For example, there are 3 languages: Java, Python, and C++, you must learn 2 of them. Then there are 3 combinations: [Java, Python], [Java, C++], and [Python, C++].

Let function  $(n, x)$  to represent the number of combinations, given  $n$  languages to choose from and  $x$  languages to learn, the formula is shown below:

$$\text{function}(n, x) = \text{function}(n-1, x) + \text{function}(n-1, x-1)$$

Write a program to solve this problem. The partially completed program is as follows. You just need to complete the base cases.

```
import java.util.Scanner;

public class Languages {

    static Scanner keyboard = new Scanner(System.in);

    public static void main(String[] args) {

        System.out.print("Enter how many programming languages are available to learn: ");
        int n = keyboard.nextInt();

        System.out.print("Enter how many programming languages that you must learn: ");
        int x = keyboard.nextInt();

        System.out.print("Combinations: " + function(n, x));

    } //end main

    public static int function (int n, int x) {

        /*
         Complete the base cases
        */

        return function(n - 1, x) + function(n - 1, x - 1);

    } //end method
} //end class
```

Sample Run Data:  $n = 6, x = 4 \rightarrow$  Combinations: 15

3. Extra Credit (10 points)

Use Java Exception Handling techniques (try ... catch) to handle exceptions for the second program.

The program should continue to accept the user input until the valid data are entered.

**Sample Run:**

```
Enter how many programming languages are available to learn: -12
java.lang.Exception: Number must be greater than 0.

Enter how many programming languages are available to learn: tw
java.util.InputMismatchException: Enter an integer.

Enter how many programming languages are available to learn: 6
It is valid.

Enter how many programming languages that you must learn: abc
java.util.InputMismatchException: Enter an integer.

Enter how many programming languages that you must learn: -4
java.lang.Exception: Number must be greater than 0.

Enter how many programming languages that you must learn: 4
It is valid.

Combinations: 15
```

**Due date: Wednesday, 10/19/22**

- To receive full credit, the assignment must be submitted by the due date. Late submissions will incur a penalty of 5% per day.
- To submit the assignment, make a zip file of the entire project and upload it to D2L. If you wrote the program with individual source files, upload them to D2L.
- For the written question, submit the answer sheet to D2L.

Each program should have a file header section. /* * Author:       Your name * Date:         Date of completion * Assignment:   Assignment # NameOfSourceCode.java * Description:   The program description */	Up to 5% deduction
Each program should be written with the appropriate form and style. Use indentation, blank line, and comments to make the source code easy to read.	Up to 5% deduction
Use Java naming convention and meaningful names to name the classes, methods, variables, constants, and other identifiers in the programs.	Up to 5% deduction
Format the output appropriately	Up to 5% deduction