

# CMPS 261: Project 1

## Due Date: Check on Moodle

### Instructions

1. You are allowed to discuss the problem and solution design with others, but the code you submit must be your own. Your solution must include the certification of authenticity as given in the course syllabus.
2. Create project in NetBeans 8.1 using Java 8
3. Name your project as P1\_<your clid>
  - a. Example: if clid = abc1234 then project name = P1\_abc1234
4. Follow the instruction on Moodle under "Submitting to Moodle".
5. Create **only one .java file** for all the problem and implement all the methods in the same class.
6. Include code in main function to test your recursive methods even if not specifically asked.

### General Requirements:

- Project submissions not including a Certification of Authenticity will not be graded until such a written and signed submission is provided by the student.
- Code, even if substantial, that fails to compile will not receive a grade higher than 25%
- Code, even if substantial, that fails to run will not receive a grade higher than 50%
- A penalty of up to 10% may be assessed for lack of required documentation, lack of descriptive identifiers, lack of indentation of blocks of code, failure to follow naming requirements of the project and associated tar archive.
- Failure to submit the complete NetBeans project may result in a penalty of up to 10%.

### Problems

- 1) (10 Points) The sum of numbers from 1 to  $n$  can be defined recursively as follows:
- a. If  $n = 1$ , the sum is 1.
  - b. for  $n > 1$ , the sum from 1 to  $n$  is  $n$  more than the sum from 1 to  $n-1$

Write an int-method named sum that accepts an int parameter,  $n$ , and recursively calculates and returns the sum of the numbers from 1 to  $n$ .

- 2) (15 points) Given  $m > n$ , Let  $\text{gcd}(m,n)$  denotes the greatest common divisor for integers  $m$  and  $n$ . The  $\text{gcd}(m,n)$  could be defined recursively as follows:
- a. If  $m\%n$  is 0,  $\text{gcd}(m,n) = n$ . Note:  $m\%n$  = the remainder left after dividing  $m$  by  $n$ . For example,  $7\%3 = 1$ ,  $11\%3 = 2$ , and  $8\%28 = 8$ .
  - b. Otherwise,  $\text{gcd}(m,n) = \text{gcd}(n, m\%n)$

Write an int-method named gcd that accepts two int parameters,  $m$  and  $n$ , and recursively calculates and returns the gcd of the two given numbers.

- 3) (15 points) Write a recursive method that converts a decimal number into a binary number as a string. Write a test program that prompts the user to enter a decimal number and displays its binary equivalent. Assume that input decimal numbers are non-negative numbers.

- 4) (15 points) The  $n$ th harmonic number is defined non-recursively as:

$$H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n}$$

Come up with a recursive definition and use it to guide you to write a method definition for a double-valued method named “harmonic” that accepts an int parameter  $n$  and recursively calculates and returns the  $n$ th harmonic number. Write a test program that displays the harmonic numbers,  $H(n)$ , for  $n = 1, 2, 3, \dots, 10$ .

- 5) (15 points) Write a recursive method to compute the following series:

$$m(i) = \frac{1}{3} + \frac{2}{5} + \frac{3}{7} + \frac{4}{9} + \frac{5}{11} + \frac{6}{13} + \cdots + \frac{i}{2i+1}$$

Write a test program that displays  $m(i)$  for  $i = 1, 2, 3, \dots, 10$ .

- 6) (15 Points) Write a recursive method that returns the largest integer in an array. Write a test program that prompts the user to enter a list of eight integers and displays the largest element.
- 7) (15 points) The “odd/even factorial” of a positive integer  $n$  is represented as  $n!!$  and is defined recursively as:
- a.  $(n)*(n-2)*(n-4)\dots\dots*(2)$  if  $n$  is even. For example,  $6!! = 6*4*2 = 48$
  - b.  $(n)*(n-2)*(n-4)\dots\dots*(5)*(3)*(1)$  if  $n$  is odd. For example,  $7!! = 7*5*3*1 = 105$ .

Come up with a recursive definition for  $n!!$  and use it to guide you to write a method definition for a method called “oddevenfact” that recursively calculates the odd/even factorial value of its single int parameter. The value returned by “oddevenfact” is a long.

### **Additional Requirements**

- [1] All output must be labeled.
- [2] Identifiers must be descriptive, i.e. must self-document. The only exception granted is in the case of a “for variable”, that is a variable created as a simple counter as in the control variable in a “for” loop statement.
- [3] Indention of all code blocks (compound statements, anything in braces), including single statements following selection or while statements, is required. NetBeans will do this fairly automatically as you type if your syntax is correct. In NetBeans, ALT-SHIFT-F will re-format a whole file if your syntax is correct.
- [4] The main “.java” file [the one with the method `public static void main(String[] args)`] of your project must contain this minimal documentation:

```
// Your Name
// Your CLID
// CMPS 261
// Program Description: description of actions of code
// Certificate of Authenticity: (Choose one of the two
following forms:)
```

I certify that the code in the method functions including method function *main* of this project are entirely my own work.

{or}

I certify that the code in the method functions including method function *main* of this project are entirely my own work., but I received some assistance from {name}. Follow this with a description of the type of assistance. (For example, if you consulted a book, and your solution incorporates ideas found in the book, give appropriate credit; that is, include a bibliographical reference.) Note: You do not have to list the text, the author of the text or the instructor's examples.

[5] Each class method must include documentation immediately above the method header in the form:

```
/* method name
 * a brief definition
 * pre-conditions
 * post-conditions
 */
```

Example:

```
/* RemoveElement
 *
 * Removes an element from a list of strings stored in an
 * array by compacting the list. First locates the list
 * element to be removed, then copies each list element
 * located at an array index larger than the array index of
 * the list element to be removed to the next lower index,
 * proceeding from lowest to highest array index.
 *
 * Pre: The array passed is populated, the item parameter
 * contains a value that is not the null string and
 * the length of the list passed is > 0.
 *
 * Post: If the item passed was found in the list, the
 * matching list element has been removed, the list
```

```
*          compacted and the length of the list reduced by 1 and
*          returned.
*/
public int RemoveElement(String[] list, String item, int len)
...
```