

## COMI 2510 Advanced Programming and Design

### Assignment 5

---

Write a method that accepts a positive integer argument and returns the sum of all the integers from 1 up to the number passed as an argument. For example, if 50 is passed as an argument, the method will return the sum of 1, 2, 3, 4, ..., 50.

Use recursion to calculate the sum.

Demonstrate the method in a program. (If you wish, you can check your work with the formula  $n(n+1)/2^*$ , but you must write a recursive method to calculate the sum.)

Hand in your class or classes.

\*This is called Gauss' trick, and here is a version of the story:

Gauss told the story that when he was a boy, the teacher ran out of stuff to teach and asked them, in the remaining time before playtime, to compute the sum of all the numbers from 1 to 20 (or similar... actually, the numbers were 1 to 40!).

Gauss thought that  $1+20$  is 21. And  $2+19$  is also 21. And this is true for all the similar pairs, of which there are 10. So... the answer is 210.

[http://www.trans4mind.com/personal\\_development/mathematics/series/airthmeticGeometricSeries.htm#mozTocId269048](http://www.trans4mind.com/personal_development/mathematics/series/airthmeticGeometricSeries.htm#mozTocId269048)

## Assignment 5 Rubric

Component		Quality			
		Exceptional	Acceptable	Amateur	Unsatisfactory
	Run-time specifications 50%	50 pts: The program meets all of the run-time specifications, with no additional unspecified functionality.*	40 pts: There is additional unspecified functionality or the program produces incorrect results in no more than 5% of the customer's tests.	25 pts: The program produces incorrect results in no more than 10% of the customer's tests.	10 pts: The program produces incorrect results in more than 10% of the customer's tests.
	Design specifications 25%	25 pts: The program is a well-designed function using recursion to solve the problem. Variable names relate to the problem. Recursive solution is elegant.	20 pts: Design uses recursion but recursion is inelegant or variable names do not relate to the problem.	12 pts: Solution is recursive but is very poorly designed, i.e. recursion an afterthought rather than central to the solution, and/or variables are poorly named.	5 pts: Design does not use recursion.
	Documentation 25%	25 pts: The program contains comments including the programmer's name and date.  Javadoc comments are included as shown in the text for all classes.  There are block comments (as many as necessary) for each distinct block of code which accurately describe what the block is accomplishing.	20 pts: The header comment is incomplete but contains name and date, and/or the block comment(s) aren't clear. Javadoc comments are missing components less than 10% of the time.	12 pts: The documentation partially meets the exceptional guidelines with several poorly written comments and/or missing comments or missing components less than 25% of the time.	5 pts: 25% or more of the comments are missing or unhelpful.

\*If you want to change the functional specifications of the program in any way, you must clear it with your customer (the instructor) in writing prior to making the changes. Include documentation of the specification changes when you submit your program.