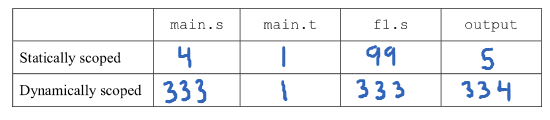
Evan Richter

PEX 1 Prelim Questions

Problem 1:



Problem 2:

1. No, because the language MR does not implement static variables. When the program is run, *tempsillyvar* will reserve a spot in memory when needed, not at once before execution.
2. If variables were given memory allocation at the beginning of execution, the variables in sillyfunc() would not be able to retain their integrity over multiple recursive loops. Once the function calls itself, the variables would be overwritten by the more recent calculations.



1. int [] john = new int[15]

int newSize = getnum(keyboard)

//user enters 20

john.incSize(newSize)

This would fail to compile in a language that uses stack-dynamic variables because it wouldn’t understand how to allocate more memory to an already declared variable. The heap-dynamic variable language, however, would be able to add on extra memory to a variable by using a pointer to a new memory chunk to make up the difference.

Problem 3:

1. &myArray
2. \*myArray
3. \*(myArray + 4)
4. It seg faults instead. When main() calls makeNewArray(myArray), the value of myArray (NULL in this case) is passed to the function. That function copied the value and modified it there, never passing anything back to main(). This means that when printf() comes along, it will (most likely) not be able to access memory location 0x0.



Problem 4:

