Homework 4: Pointers and Memory Diagrams

CSCI 60
Krehbiel

Due 11:59pm Wednesday, October 22, 2025

Overview. This mini-homework is like a self-quiz that you have 48 hours to work through. Using your computer to check your work is allowed and even encouraged!

Download `selfquiz.cpp` from and create a drawing that describes the state of memory (both the call stack and the heap) at every point in the code where a comment says `// DIAGRAM X`. Please number your diagrams to make it easy to read, and emulate our memory diagrams from class as much as possible. In particular, there is no need to write literal memory addresses; pointers represented as arrows are much preferred. Blocks of memory on the stack should all have variable names associated with them because these are all local variables declared with a name, and blocks of memory on the heap should not because they are allocated anonymously by `new`, which only reports their address. Each memory diagram should illustrate *all* memory on the stack (including local variables in `main` and every other function that has not yet terminated) and *all* memory allocated on the heap (including any orphaned memory and *excluding* memory that has been released).

Feel free to draw these diagrams with pencil and paper, a tablet, a powerpoint-like program, or whatever you prefer. Make sure your diagrams are clear and easy for me to read, but don't worry about spending lots of time making them look pretty; I'm looking for content that shows you understand the new memory concepts we've been talking about! In a perfect world, you'll be able to draw everything correctly without running the code at all – think about how this might be tested on a quiz or midterm! – but realistically, it would be good to compare your diagrams to the corresponding `cout` statements periodically to make sure you fix any errors before they propagate. Feel free to add additional `cout` statements if you want more details about exactly what's going on. When you're happy with your diagrams, upload them as pdfs or photos to Gradescope.