High Level Look at OS and Memory
The operating system allocates space for each program.

The program space is divided into 4 segments:

- **Heap** — holds objects declared with the “new” operator
- **Stack** — holds local variables
- **Data** — holds global and static variables
- **Code** — holds program code

The heap grows downwards.
The stack grows upwards.
const double PI = 3.14;

int multiply(int a, int b);
int add(int a, int b);

int main() {
    int *result = new int();
    *result = multiply(8, 4);
    cout << "8 * 4 = " << *result;
}

int multiply(int a, int b) {
    int temp = 0;
    for (int i = 0; i < b; ++i)
        temp = add(temp, a);
    return temp;
}

int add(int a, int b) {
    return a + b;
}
Processes and Threads

- When you run your program, the operating system creates a process that executes your program.
Processes and Threads

- There are multiple processes running on your computer at any given time.
Processes and Threads

- Processes are put on a queue and the operating system’s scheduler decides when a process gets put into the CPU core.
- The scheduler swaps processes out of the CPU core after it’s time quota has been reached.
- If you have a CPU with 1 core, it creates the illusion of multi-tasking.
Processes and Threads

- Processes have separate memory
- Threads are “mini” processes
- Threads run inside processes and share the processes memory space
- Threads allow processes implement concurrent programming
  
  - 1 thread running main function
  - 2 threads running timers
  - 1 thread maintaining connection to a server
Cache

CPU/Memory performance

![Graph showing the performance of CPUs and memories over time, with a gap between the two curves indicating the performance gap.](image)
Cache

- dictionary.com
  - a hiding place, especially one in the ground, for ammunition, food, treasures, etc.

- wikipedia.com
  - In computer science, a cache is a component that transparently stores data so that future requests for that data can be served faster
Cache

Diagram of computer memory hierarchy:

- CPU
- L1 Cache
- L2 Cache
- RAM
- Disk
Cache

- Caches allow very fast retrieval of data if that data is stored in the cache
  - cache hit – data is in cache
  - cache miss – data is not in cache
- When a miss occurs, data is searched for in the next memory level
- The retrieved data is placed into the lower level memory so next access will (hopefully) be a hit
Any order of i, j, and k will give you the correct result, but the runtimes may differ.

runtime(i, j, k) > 2 * runtime(i, k, j)

You get more cache misses with i, j, k order because you are searching addresses spaced farther apart, thus needed data is getting swapped out of the cache often.

Matrix Multiplication
for (int i = 0; i < n; ++i)
    for (int j = 0; j < n; ++j)
        for (int k = 0; k < n; ++k)
            c[i][j] += a[i][k] * b[k][j]
Question Time!

Win a bag of cookies!
Question 1

- Implement a `MyQueue` class which implements a queue using two stacks.
Question 2

- Given an image represented by an \( N \times N \) matrix, where each pixel in the image is 4 bytes, write a method to rotate the image by 90 degrees. Can you do that in place?
Write a function to swap a number in place without temporary variables.
You are trying to cook an egg for exactly fifteen minutes, but instead of a timer, you are given two ropes which burn for exactly 1 hour each. The ropes, however, are of uneven densities — i.e., half the rope length-wise might take only 2 minutes to burn.
Solutions

...are not provided.
Questions Taken From…

CRACKING THE CODING INTERVIEW

Full Candidate Interview

Gayle Laakmann McDowell
Founder / CEO, CareerCup.com