CHEET SHEET

Summary

- Dynamic memory allocation is a process of reserving memory on-the-fly with a heap using malloc().
- The heap is part of the process's memory that begins as unused and unallocated.
- Allocated blocks are used, while unallocated blocks are free.
- Design principles for a dynamic memory allocator include aligning blocks, adding padding, and using metadata.
- Goals for an effective dynamic memory allocator include time-efficiency, space-efficiency, robustness, and locality.
- Strategies for finding free blocks include First Fit, Next Fit, and Best Fit.
- Fragmentation can occur due to internal (overhead) and external (order of allocation/freeing) issues.
- Free and used block integrity is maintained by including block size and used bit in metadata.
- Strategies for maintaining and locating free blocks include Implicit List, Explicit List, Segregated List, and Sorted by Size.
- Coalescing neighboring free blocks is important to make the best use of memory.
- Garbage collection is used in languages like Java and Python to free unused memory.
- The Mark-and-sweep algorithm is a common garbage collection method.

For Next Class:

- Finish reading 9.9-9.10.
- Start reading 8.1-8.4.
- Project 2 will be announced towards the end of the week.