a. caching - *storing data in cache memory. Quick accessibility*
b. locality - *when same value or related storage locations accessed frequently.*
c. DMA – Direct Memory Access – *allows certain hardware to access system memory independent from the cpu*
d. Conflict miss - failed attempt to read/write data in the cache – *could have been avoided had the cache not evicted an earlier entry.*
e. Capacity miss – *misses occur due to the finite size of the cache. Because its always full new entrys will overwrite older ones.*
f. Direct-mapped cache – *when each block from main memory has only one place it can appear in the cache. Allows simple and fast speculative execution.*
g. Set-associative cache - *between direct-mapped and fully associative caches. Each line is divided into sets and the middle bits of the address determine the set it is placed.*
h. Write-through caching - *disk or memory cache that supports caching or writing. After writing if a read is called, the read performance is improved because its already in high-speed cache.*

2. On a modern CPU, if you had the choice of spending 100K instructions or waiting for one disk seek to complete (assuming an average 7200RPM 3.5" drive), which would you choose?
100K instructions are still faster then then one disk seek.

3. Briefly describe the process of reading a bit from DRAM.
Requires reading a row from the memory array into the row buffer and writing it back unchanged.

4. What is the net effect of the cache hierarchy?
The cache hierarchy is a way for multiple cores to have shared cache.

5. Why would register usage be controlled by the compiler, but cache usage controlled by the CPU hardware?
Because not all variables are in use the compiler must decide how to allocate these variablaes to a small, finite set/not all processors will share access to the same memory.

6. Assume you are specifying the hardware for a program randomly accessing a large amount (think much larger than a computer's average RAM size). Which would be a better way to spend your budget – more cache, or more RAM?
Ram would have a larger impact on performance because cache sizes are small.

7. Discuss the relative merits of write-back vs. write-through caching.
Write through – data written to cache and disk (large write time dropping performance)
write back – data is only written to cache until cache is full, after full it'll write to HD
cache is power sensitive, so if power is lost the data in cache is gone.

8. Why did the blocked array multiply exhibit better caching performance?
Block multiplication reuses more data then regular array multiplication. This allows cache to be used more effectively

9. Discuss your decision process when choosing between a faster dual-core chip (with half the cache of the quad-core) or a slower quad-core chip for a given application.
Dealing with a small data size cache speeds it up more. So quad core works better.
Larger data sets dependent on processor speed

10. Applications are usually classed as CPU-bound or I/O-bound. How would you figure out which class to put a particular application end?
Cpu-bound – run themselves with little to no input from user
IO-bound – dependent on user input, which is primary cause for slowness