Review for Final Exam

Classes and Objects:

You should know the difference between a class and an object. If I give you a class definition, you should be able to pick out the variables and the methods in it.

Suppose I have a class definition called “Shooter.” (This means the constructor method is also called Shooter.) How would I create an object called player1, of type Shooter? Let's say the class definition has a method called hitshot(). How would I call the hitshot() method of player1 – what operator would I use?

Binary Arithmetic and Boolean Algebra:

You should know how to convert from binary to decimal and vice-versa. How many bits would I need to represent n symbols? What is overflow, and give an example of it? You should know to add two binary numbers and design logical circuits. The kinds of questions in Lab 6 are quite likely to appear on the exam.

Computer Architecture:

What is the difference between hardware and software, and what are some examples? Which unit of a computer contains the circuitry to execute machine instructions? What are the 3 kinds of machine instructions? Mention a difference between RAM and ROM. Give 2 examples of secondary memory. Amongst registers, cache and main memory, which is the fastest to access and use? What is a compiler?

Algorithmic Complexity:

What is the difference b/w tractable, non-tractable and non-computable problems in CS? Give examples.

Given a number n, what does it mean to say that an algorithm takes $O(n^2)$ time? What is a sequential/linear search, and what is its running time?
for a list of n items? What is a binary search, and what is its running time for a sorted list of n items? Does binary search work on unsorted lists? Is the difference in running time b/w linear and binary search going to be more apparent when n is small or when n is large?

Understand the basic idea of selection sort, insertion sort and merge sort. Also know the running times of each in the best and worse cases.

Just having faster processors or parallel processors does not solve intractable problems. Why is this? What is one of the biggest open questions in CS?

**AI:**

Are computers currently better at simple tasks or complex tasks, when compared to humans? What is the Turing Test?

**Recursion:**

What is recursion? What is the difference between the base case and the recursive case? Write some code to recursively compute n! or the n-th term in the Fibonacci series. Can every recursive algorithm be written iteratively?

Understand the examples at the end of Lecture 22, involving the code snippets. Review the questions from Lab 8.

**Security:**

What do confidentiality, integrity and availability mean? Give examples of attacks on them. What do these terms mean: encryption, decryption, cipher text? What are the 2 kinds of encryption algorithms? Briefly, what is the difference between the 2 kinds?