CSE120 Sp17 Final Review

Note: You can bring two double-sided, letter-sized (8.5”×11”) sheets of handwritten notes to the exam.

Lecture Main Ideas

Computer Science Principles has covered two kinds of information:

1) Ideas and details concerning programming, such as arrays and Strings
2) Ideas about computational topics, such as privacy and artificial intelligence

The final covers both of these kinds of information, including all the lecture material so far:

15. **Privacy:** The right of people to choose freely what circumstances and to what extent they will reveal themselves, their attitude, and their behavior to others. Technology and the Internet produce tons of data that can be used to invade your privacy. *Cybercasing* is using online data and services to enable “real-world” crimes that would not otherwise be possible. One method of doing this involves *multimedia retrieval*. We examined the problems with geotagging data.

16. **Recursion:** An algorithm can accomplish a task by calling itself on some part of the task! Anything written recursively can be rewritten iteratively, but recursion is often a more natural solution to *self-similar* problems (problems that have subproblems that are similar to the original problem). Follows the normal rules of function calling and returning – each recursive call creates a new environment that doesn’t interfere with the others. Need a *base case* to stop infinite recursion and the *recursive case* will recurse on “smaller” problems.

18. **Proofs and Computation:** You can actually use theory to prove a lot of interesting results related to computation without ever running anything! In particular, no matter how powerful of a machine you use, there are certain things that you *cannot compute* (*i.e.* computers are not all-powerful)! The example shown is known as the Halting Problem.

19. **Mid-Quarter Review:** Strings are collections of characters (*i.e.* words and sentences). They can be combined using the concatenation operator (+). You can retrieve an individual character (`str.charAt(i)`) or the String length (`str.length()`).

20. **Puzzle App:** Define clickable regions on canvas using conditionals and arithmetic. Store game “state” using variables and arrays. Use some numerical encoding to represent board game states. *Functional abstraction* to manage complexity and *refactoring* to update old code as you go.

21. **Security:** Use a *security mindset* to think like an attacker in order to improve the security and privacy of technologies: analyze the different assets and how they might be attacked, then build new defenses for any vulnerabilities that are found. While sending information over the Internet (and through many machines), we rely on *encryption* so that others can’t decipher it, though we still have to trust certain *certificates* and sources.

22. **Animation:** You won't be tested on animation.

24. **Artificial Intelligence:** Difficult to define, but could be the science of making machines that (1) think like people, (2) act like people, (3) think rationally, or (4) act rationally. Most modern AIs are specialized to a specific task (artificial *narrow* intelligence), rely on probability and statistics to make decisions, and are applied to a wide variety of different applications. The new hot field is *machine learning* and make future decisions based off of extensive training set data. AI raises a ton of ethical issues as we place more and more decision-making in the hands of machines.

25. **Digital Distribution:** The ability to copy digital information instantaneously and without loss has a huge legal, cultural, and economic impact on content creators. *Copyright* protects the rights of the creator to the use and distribution of his/her *intellectual property*. If the copyright holder chooses to, he/she can distribute work fully under an *open-source license* (typically applies to software) or with some restrictions using a *Creative Commons license*.

26. **Computing in the Developing World:** Computing and technology can be a huge boon to people and areas that don't typically have access to them. However, when designing tech for people/users, you have to take into account their social and cultural contexts as well as environment. Technology and computing that we take for granted can, if adapted correctly, greatly improve the lives of people in the developing world.

27. **Big Data:** Most fields of study (science & engineering, especially) now generate huge amounts of data that hold the promise to accelerate discovery. We need to develop new tools to both store and analyze that data, such as *database management systems* (most of which now run in the cloud). Want users to be able to perform data analysis using queries in a declarative programming language (*e.g.* SQL) and custom scripts without needing to know the details of the query optimizations that occur "under the hood."

29. **Social Implications of Computing:** Because computers and access to the Internet are now ubiquitous, they have fundamentally changed the way we think, act, and work. The abundance of information allows us to choose what we believe in because we often end up in biased *filter bubbles* and seek out opinions that corroborate our own. This can either polarize people or make them more tolerant. The Internet (and social media) is great for forming/maintaining *weak ties* that are useful for organizing and raising awareness, but often lack the *strong ties* that lead to actions that affect real change, leading to phenomena such as *slacktivism*. Finally, computing and automation are reshaping the job market and labor pool in ways that will have profound effects on the future of education, work/leisure, and economics.

Practice Questions

Completing these questions will help you prepare for the final, but please note that this list does not cover all material that may be on the final, but instead is a guide to help you review. Completing these on a separate piece of paper is suggested, as extra space is not provided here. It is highly recommended to write out code on *paper* first to give you practice for the exam. Then you can type your answers into Processing to test and verify. Feel free to work together!

Questions will be based on the lectures, readings, and assignments:

1. True or false? Looping is necessary for complex programs. Briefly explain.

2. Briefly explain the benefit of using *parameters*. What is the *variable scope* of parameters?

3. Briefly explain the benefit of using *arrays*. Name a few examples of array usage in this class.

4. Calculating exam statistics takes time! If was assume that all of the scores for the final are stored in an array, *describe in words* (not code) algorithms to calculate the MEAN (average) and MEDIAN (middle score). Your answers should reference the array.

5. We want to use an array of colors of length 6. Write Processing *code* below that declares such an array (use any name of your choosing) and initializes them to different shades of blue evenly spaced from black to totally blue.

6. Write a Processing *function* called `grid()` that draws a grid of rectangles of width 50 and height 20. The user should be able to specify the coordinate of the upper-left corner of the grid as well as the number of rows and columns of rectangles.

7. Write a Processing *function* called `hasX()` that returns `true` if the String that is passed to it contains the character `'x'` in it and `false` otherwise.

8. Write a Processing *program* that draws green squares of size 50 along the diagonal of your drawing canvas, starting from the upper-left and going towards the lower-right. The squares should touch neighboring squares at their corners.

9. Write a Processing *program* that draws a black line of length 100 that is centered in the middle of your drawing canvas. This line should rotate clockwise at a constant rate.

10. Write a `keyPressed()` function that detects the number keys (0-9) and prints the associated symbols found above them on the keyboard (*e.g.* 1 → !, 2 → @, 3 → #) to the console.

11. Examine the recursive function below. What gets printed to the console when we call `mystery(4)`? What happens when we call `mystery(-1)`?

```
void mystery(int n) {
  if( n != 0 ) {
    println( (3*n+4)%5 );
    mystery(n-1);
  }
}
```

12. [more complex] Write a Processing *program* that constantly draws colored lines from the position of the last mouse click to the current mouse position. Initially, the "last mouse click position" should be (0,0). Each time the mouse is pressed, the drawing canvas should be cleared and the line color should change (use a rotation of red → green → blue). New lines will be drawn from this new position to the current mouse position.

13. Give an example of a *multimedia retrieval* request that you wish you could search for.

14. Name the two components that are necessary in a recursive function and briefly explain what each one does.

15. Most modern video game consoles not only play games and movies, but have cameras, microphones, and online stores. Thinking about the security of such a system, name two different assets that an attacker might go after and what the attacker might do with that stolen information.

16. A computer that can pass the Turing test accomplishes which goal of AI? (Circle one)

    Think like people   Act like people   Think rationally   Act rationally

17. Give an example of a digital rights management (DRM) technology. Briefly explain how it is intended to upload copyright.

18. Name a technology that would greatly benefit a village in a developing country where supplies (medicine, food, books, etc.) are delivered infrequently (*e.g.* once every 3 months). Also name a major design consideration (*e.g.* power, durability, portability) for that technology in this situation.

19. Name two benefits of using a database management system in the cloud versus writing custom scripts on your local machine.

20. Briefly explain what the Filter Bubble effect is and why it is problematic.

21. Which is usually larger in size: a text file or a video file? Briefly explain why.