CSE120 Sp17 Midterm Review

Note: You can bring one double-sided, letter-sized (8.5"×11") sheet 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 variables and for-loops
2) Ideas about computational topics, such as abstraction and binary

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

1. **Abstraction:** A technique for dealing with complexity by *generalization* and/or *removal of details*. In computer science, we break complex problems into smaller, more manageable, and reusable tasks.

2. **Lightbot and Functions:** Lightbot 2.0 seemed like a game, but it was also *programming*: the act of preparing *instructions* that an *agent* follows to achieve a specific *goal*. *Functional abstraction* is the act of identifying a sequence of operations that achieve a significant goal and giving them an identifiable *name*.

3. **Binary:** You can represent anything countable using just 0's and 1's (bits and bytes)! How the binary number system works, including conversion into other bases (decimal and hexadecimal).

4. **Processing Intro:** Processing is text-based, so we write down instructions and the computer follows them when we execute a program. Learned some basic syntax, including semi-colons, *colors* (RGB), and *shapes*.

5. **Variables & Datatypes:** A *variable* is a piece of your program that holds a value. You *declare* a variable by specifying its name and *datatype* (`int`, `float`, `color`, `boolean`, etc.), but then need to *initialize* the variable to some initial value before using it. You can modify the value in a variable throughout your program. Also learned about the `min()` and `max()` functions.

6. **Functions in Processing:** *Functions* are defined once, but called as many times as desired. A function declaration includes a *return type* (datatype), name, and list of *parameters* (internal variables). Parameters allow different calls to a function to vary by passing the *argument* values of the function call into the parameter variables, which are used in the function *body*. Variables only exist in the *variable scope* in which they were declared (local vs. global).

7. **Expressions & Control Flow:** An *expression* is a combination of values, variables, operators, and functions that produce another value. Operators are built-in "functions" that use special symbols. Control flow is the order in which instructions are executed. This is normally sequential from top-to-bottom, but is changed by function calls, *conditional statements*, and *loops*. If-statements allow for decision-making and for-loops allow for sequences of repetitions. Also learned about the modulus (%) operator.

8. **Basic Input & Output:** *System variables* are special variables that hold values related to the state of the program, including user input (`mouseX`, `mouseY`, `mousePressed`, `key`, `keyPressed`). Text output can be printed to the console or the drawing canvas. Each key press is represented by the ASCII encoding of the associated *character*.

9. **Algorithms:** A *computational problem* specifies a desired input/output relationship. An *algorithm* describes a procedure to solve a computational problem. An *implementation* is an algorithm expressed in a way that a computer/agent can execute it. There are many different possible algorithms for every computational problem.

10. **Nested Loops & Arrays:** *Nested loops* are loops inside of other loops and allow for changing multiple things in different ways at the same time. *Arrays* are structures that store many values of the same datatype, accessed using the array name along with the *index* of the *element* you want.

11. **Algorithmic Complexity I:** Needs metrics to compare different algorithms. One common metric is computation time, often measured by the worst-case scenario. *Orders of growth* allow us to characterize algorithms by how their runtimes vary by input size.

12. **The Internet:** A *network* is a group of computing devices connected together. *The Internet* is the largest network of networks on the planet and was built to be *decentralized* and using *open standards*. Specific devices can be reached using the *domain name system* and *IP addresses*.

13. **Computers:** A computer has five main components (*control*, *datapath*, *memory*, *input*, and *output*) and is a combination of hardware and software, managed by an *operating system*. The CPU interprets instructions using a fetch and execute cycle, which is a mechanical and deterministic method of performing instructions in order.

Practice Questions

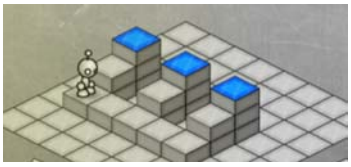
Completing these questions will help you prepare for the midterm, please note that this list does not cover all material that may be on the midterm, but instead is a guide to help you review. Completing these on a separate piece of paper is suggested, as extra space is not provided here. It is highly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

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

1. Lightbot questions
   a. How does the Lightbot keep track of where it is in the program?

   b. Lightbot (and computers) can do only a small amount of instructions; how do programmers avoid the tedium of using these instructions all the time?

   c. If F_1 is a Lightbot function that contains: `Left, Jump, Jump, Left`. Write an F_2 so the solution for the problem at right is `3:(`F_2`)`.



2. Abstraction is recognizing a concept (for us) in a sequence of operations and giving it a name; why do we bother to abstract?

3. Give two ways in which the functions of Processing are "more powerful" than the functions of Lightbot 2.0 – what can a Processing function do that a Lightbot 2.0 function can't?

4. What is the difference between an *active* and a *static* program in Processing? Provide some examples in your answer.

5. Is the following line of code correct, incorrect, or unknown? Explain your choice.

```
line(150, 150, mousex, mousey);
```

6. Describe the following background colors

```
a. background(255, 0, 0);
b. background(64);
c. background(0, 0, 64);
```

7. Define the term *variable*.

8. State the datatype of each of the values below:
   a. `3.14`
   b. `color(0)`
   c. `{1, 2, 3}`
   d. `-3`
   e. `'x'`
   f. `true`

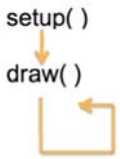
9. State the datatype of each of the values returned by the system variables below:
   a. `mouseX`
   b. `width`
   c. `key`
   d. `keyPressed`
   e. `frameCount`

10. In your own words, describe what the statement below does.

```
int ra = 200;
```

11. What are the possible values that a `boolean` variable can hold?

12. Explain the diagram below in your own words.



13. Explain what the `min()` function does.

14. How many parameters does the function header below have?

```
void ninja(int x_pos, int y_pos, color goggles)
```

15. Write a piece of code that will display a circle 20 pixels wide that flashes random colors.

16. If I wanted to write a program that draws 4 identical faces on the screen, explain in words (not code) the best way to go about this. What kind of programming constructs should you use?

17. In the code below, which variable names are *parameters*? Which variable names are *arguments*?

```
color gray (int shade, int number) {
      record = max(number, record);
      return color(shade+2, shade+2, shade+2);
}
```

18. Do the two statements below mean the same thing? Explain your answer.

```
seconds = seconds + 1;
seconds + 1 = seconds;
```

19. List all of the datatypes that we have used so far in Processing.

20. Given the for-loop below, draw the result. Label at least one point per shape.

```
for (int i = 0; i < 4; i = i + 1) {
      rect(50, 50+20*i, 10, 10);
}
```

21. If the 4 in the code above (Question 20) is changed to a 5, what will change in the drawing?

22. If `dog(xpos, ypos)` is a function drawing a dog at the position xpos,ypos, write a for-loop to draw ten dogs in a row, with a dog every 100 pixels in the horizontal direction.

23. Name one benefit of digitizing information.

24. Name 5 uses of a computer today, which are not connected to this class.

25. What is an Internet packet?

26. What is a "byte"?

27. The binary number `10 0101` represents what decimal number?

28. What are all the possible three-bit binary sequences?

29. Name or describe a famous algorithm that you use on a regular basis.