

CSE120 Wi19 Final Review

Practice Question Solutions

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

False. Many loops can be explicitly written out as individual statements and others can be replaced with combinations of other programming constructs like conditionals and functions.

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

Parameters allow us to make changes between different calls to the same function, which generalizes the task at hand and makes the function more widely useful. Parameters are variables that exist only *within* the function in which they are declared.

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

Arrays allow us to group multiple variables of the same type under the array name and declare/initialize them all in one go instead of individually. Differentiating these variables by index (integer) makes accessing arrays via loops very convenient.

Examples: position and color of Elli's segments, game state of Tic-Tac-Toe and keypad, store data loaded from files in Bday Viz, the `pixels []` array that holds the color data of the drawing canvas, plus many more!

4. Calculating exam statistics takes time! If we 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.

MEAN: Keep a sum variable that starts at 0. Go through all indices of the array and add the values to the sum variable. Return the sum variable divided by the length of the array.

MEDIAN: Sort the array, then return either return the value in the middle index if the array has an odd length or return the average of the middle two indices if the array has an even length.

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.

```
color[] myColorArray = new color[6];
for ( int i = 0; i < 6; i = i + 1 ) {
    myColorArray[i] = color(0,0,255/5*i);
}
// Alternatively, could have explicitly initialized:
color[] myColorArray = {color(0), color(0,0,51), color(0,0,102),
    color(0,0,153), color(0,0,204), color(0,0,255)};
```

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.

```
void grid( float x, float y, int rows, int cols) {
    for ( int i = 0; i < rows; i = i + 1 ) {
        for ( int j = 0; j < cols; j = j + 1 ) {
            rect( x + 50*j, y + 20*i, 50, 20 );
        }
    }
}
```

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.

```
boolean hasX( String s ) {
    for ( int i = 0; i < s.length(); i = i + 1 ) {
        if ( s.charAt(i) == 'x' ) {
            return true;
        }
    }
    return false;
}
```

8. Write a Processing `draw()` function 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.

```
void draw() {
    noStroke(); // this is optional
    fill(0,255,0);
    for ( int i = 0; i < min( width, height ); i = i + 50 ) {
        rect( i, i, 50, 50 );
    }
}
```

9. Write a Processing `draw()` function that draws a thick, horizontal, black line of length 100 that is centered in the middle of your drawing canvas. This line should follow your mouse vertically (but not horizontally).

```
void draw() {
    background(255); // background color of your choice
    stroke(0); // black stroke
    strokeWeight(8); // any large-ish number works here
    line( width/2-50, mouseY, width/2+50, mouseY );
}
```

10. What gets printed to the console when the following program is run?

```
String soda = "Pepper";
String pop = "Coke";
println("Is it " + soda.equals(pop) + " that Dr " + soda + " has " +
2 + soda.length() + " flavors?");

"Is it false that Dr Pepper has 26 flavors?"
```

11. 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.

```
// The following would work using chars instead of Strings in
// println() (e.g. '!' instead of "!").
// Also would work with separate if-statements instead of
// if-elseif.

void keyPressed() {
  if ( key == '1' ) {
    println("!");
  } else if ( key == '2' ) {
    println("@");
  } else if ( key == '3' ) {
    println("#");
  } else if ( key == '4' ) {
    println("$");
  } else if ( key == '5' ) {
    println("%");
  } else if ( key == '6' ) {
    println("^");
  } else if ( key == '7' ) {
    println("&");
  } else if ( key == '8' ) {
    println("*");
  } else if ( key == '9' ) {
    println("(");
  } else if ( key == '0' ) {
    println(")");
  }
}
```

12. Complete the following function that prints a triangle of asterisk characters of a specified height to the console. Some examples:

`draw_triangle(3)` should print out:

```
*
***
*****
```

`draw_triangle(4)` should print out:

```
*
***
*****
*****
```

```

void draw_triangle(int height) {
  for ( int i = 0; i < height; i = i + 1) {
    for ( int j = 0; j < 2*i + 1; j = j + 1 ) {
      print( "*" );
    }
    println();    // starts a new line
  }
}

```

13. 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.

```

float lastX = 0;
float lastY = 0;
int currC = 0;    // current color
color[] drawC = {color(255,0,0),color(0,255,0),color(0,0,255)};

void setup() {
  size(500,500);    // whatever size you want
  background(200); // whatever background color you want
}

void draw() {
  stroke( drawC[currC] ); // instead of array access,
                        // could be 3 if-statements instead
  line( lastX, lastY, mouseX, mouseY );
}

void mousePressed() {
  background(200); // clear screen
  lastX = mouseX;
  lastY = mouseY;
  currC = (currC + 1) % 3; // rotate to next color
}

```

14. Fill in the blanks to complete the function `twoSum()` below, which takes in a non-empty array and a value and returns whether or not there exist two *different* elements (*i.e.* different locations, though values could be the same) in the array that sum to the specified value. Examples:

`arr: {1,5,7,2,3}, val: 10` → `twoSum(arr, val)` returns `true` because $7 + 3 = 10$

`arr: {2,4,9,2,0}, val: 7` → `twoSum(arr, val)` returns `false`

```
boolean twoSum( int[] arr, int val ) {
    for ( int i = 0; i < arr.length; i = i + 1 ) {
        for ( int j = 0; j < arr.length; j = j + 1 ) {
            if ( (i != j) && ( arr[i]+arr[j] == val ) ) {
                return true;
            }
        }
    }
    return false;
}
```

15. Fill in the blanks below to complete the function `arrayConcat()`. This function takes in two non-empty integer arrays and returns a single integer array that is the two arrays concatenated (put together). `arr1` should be *before* `arr2` in the returned array. This could be used to help Elli grow after eating an apple! Examples:

`arr1: {1,1}, arr2: {2,2,2}` → return `{1,1,2,2,2}`

`arr1: {80}, arr2: {100,120,140,160}` → return `{80,100,120,140,160}`

```
int[] arrayConcat(int[] arr1, int[] arr2) {
    int newLength = arr1.length + arr2.length;
    int[] concatArr = new int[newLength];
    //copy over arr1 into front of concatArr
    for ( int i = 0; i < arr1.length; i = i + 1 ){
        concatArr[ i ] = arr1[ i ];
    }
    for ( int i = 0; i < arr2.length; i = i + 1 ){
        // how many spots have already been filled?
        concatArr[ arr1.length+i ] = arr2[i];
    }
    return concatArr;
}
```

16. Fill in the blanks below to complete the function `charsToString()`. This function should take in a character array and return a `String` that contains the characters in the array in order. Examples:

`arr: {'h','e','l','l','o'}` → `charsToString(arr)` returns `"hello"`

`arr: {'I',' ','<','3',' ','U'}` → `charsToString(arr)` returns `"I <3 U"`

```
String charsToString( char[] arr) {
    String output = "";
    for ( int i = 0; i < arr.length; i = i + 1 ) {
        output = output + arr[i];
    }
    return output;
}
```

17. 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.

[many possible answers here]

Microphone – listen in to conversations in your house or the audio as you play a game

Camera – look in at your house (who's there?)

Online stores – credit card information, linked accounts, purchasing power using your account

18. 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

Needs to trick a human into thinking it's a person by chatting with them.

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

The Filter Bubble effect is where you are exposed primarily (or entirely) to information and opinions that confirm your preexisting beliefs. This is problematic because you aren't getting the cross-cutting exposure necessary to make informed conclusions and decisions.

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

Video file. Images are stored as pixel data and videos cycle through many images per second.

21. What do the acronyms **HCI** and **UX** stand for? Why are these important?

HCI is human-computer interaction and UX is user experience. These are important to improve the usability, accessibility, and pleasure for when users interact with your program/product.

22. Name one machine learning/artificial intelligence/game theory **algorithm** mentioned in the lecture AND a **computation problem** that that algorithm can be applied to.

Minimax algorithm – solves games like Candy Crush and Tic-Tac-Toe

Neural network (I know, not an algorithm, but topical enough) – computer vision

Automatic speech recognition – convert audio to text

... and many more