University of New Mexico
CS 105-JavaScript Final Examination
Spring 2016

# Clearly print (not sign) your name:

_____

If you are sick or otherwise temporarily impaired, then request that you take the exam at a later time. This examination is an opportunity for you to demonstrate the great things you know about the topic. It is not intended as a test of how well you can perform while sick or otherwise unduly stressed.

You may use one page of handwritten notes (both sides) and a non-electronic dictionary. **No** phones, calculators, tablets, laptops or other electronics are allowed. Write your answers on the exam. You may use extra scratch paper (blank, ruled or graph).

When a question asks: "What is displayed on the canvas?" then, you need only answer what is actually displayed (there is no need to give a complete sentence). For example, if the question says:

> The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What number is displayed?
>
> ```
> var z = 2;
> z = z + 1;
> fill(0, 0, 0);
> textSize(12);
> text(z, 30, 30);
> ```

Then the full-credit answer is simply:

```
3
```

Usually, a box around the answer is not necessary; however, if you think your answer might be difficult to distinguish from any intermediate scratch work you might have written on the page, then draw a box around the part you want graded as your answer.

If your answer is correct, then any scratch work is ignored. However, if your answer is wrong, then any scratch work you provide may be useful for awarding partial credit. If you choose to use extra scratch paper, you may keep it or turn it in with your exam. If you choose to turn in extra scratch paper, then *print your name* at the top of each page and label scratch work with the question number to which it pertains.

This exam is designed for 2 hour period.

1) The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What is displayed in the console?

```
1)  var dara = 25;
2)  var x = (2 + 5)*(3 + 3);
3)  var y = 1 + 2 + 3 * 10;
4)  var z = 23 % 5;
5)  console.log(x +", " + y + ", " + z);
```

2) The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What is displayed in the console?

```
1) var dara = 25;
2) var rowan = 3;
3) rowan = rowan + 1;
4) if (rowan < dara)
5) {
6)    rowan = rowan * 2;
7) }
8) if (rowan < 8)
9) {
10)   rowan = rowan - 5;
11) }
12) if (rowan > 5)
13) {
14)   dara = rowan + 2;
15) }
16) console.log("dara="+dara);
17) console.log("rowan="+rowan);
```

3) The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What is displayed in the console?

```
1)  var dara  = 5;
2)  var rowan = 3;
3)  rowan = rowan + 5;
4)  if (rowan < dara)
5)  {
6)     rowan = rowan * 2;
7)  }
8)  else if (rowan < 100)
9)  {
10)    rowan = rowan + 10;
11) }
12) else if (rowan > 0)
13) {
14)    rowan = rowan + 200;
15) }
16) else if (dara > 0)
17) {
18)    dara = dara + 300;
19) }
20) console.log("dara="+dara);
21) console.log("rowan="+rowan);
```

4) The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What is displayed in the console?

```
1)  var msg = "kathy is now ";
2)  var kathy = 1;
3)  while (kathy < 25)
4)  {
5)      console.log(msg+kathy);
6)      kathy += 5;
7)  }
8)  console.log("***Done*** " + msg + kathy);
```

5) The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What is displayed in the console?

```
1)  var a = 1;
2)  var b = 1;
3)  var c = 1;
4)  while (c < 25)
5)  {
6)      a = b;
7)      b = c;
8)      c = a + b;
9)      console.log(a + ", " + b + ", " +c);
10) }
11) console.log("c=" + c);
```

6) The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What is displayed in the console?  Note: When `console.log`(*str*) is called with the same value of *str* as its last call, many browsers will display *str* only once preceded by a count of the number of times that same line of output was repeated. You get full credit whether you give your answer by replacing repeated lines with a count OR of you display each output string (even repeated output) on different lines.

```
1)  var a = 1;
2)  var x = 1;
3)  while (x < 8)
4)  {
5)      if (x > 5) console.log("Ray");
6)      else if (x > 3) console.log("Finn");
7)      else console.log("Maz");
8)      x = x + 1;
9)  }
10) console.log("x=" + x);
```

7) The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What is displayed in the console?

```
1) var a = 1;
2) var myList = ["Poe", "BB-8", "Luke"];
3) console.log(myList);
4) console.log(myList[1]);
5) console.log(myList[2]);
```

8) The JavaScript / Processing.js code below runs without errors and displays a number on the HTML canvas. What is displayed in the console?

```
1) var numList = [];
2) var x = 1;
3) while (x < 8)
4) {
5)     numList.push(x);
6)     console.log(x + ": " + numList);
7)     x = x + 2;
8) }
9) console.log("Done: " + numList);
```

9) The JavaScript / Processing.js code below runs without errors and draws an image on a 200x200 pixel HTML canvas. Sketch a reasonable approximation of what is in the screen capture of the canvas below.

```
1) background(220, 220, 255); //red, green, blue
2) stroke(0,0,0);
3) var i = 0;
4) while (i<=200)
5) {
6)     line(0, 0, i, 200-i);    //x1,y1,x2,y2
7)     i += 10;
8) }
```
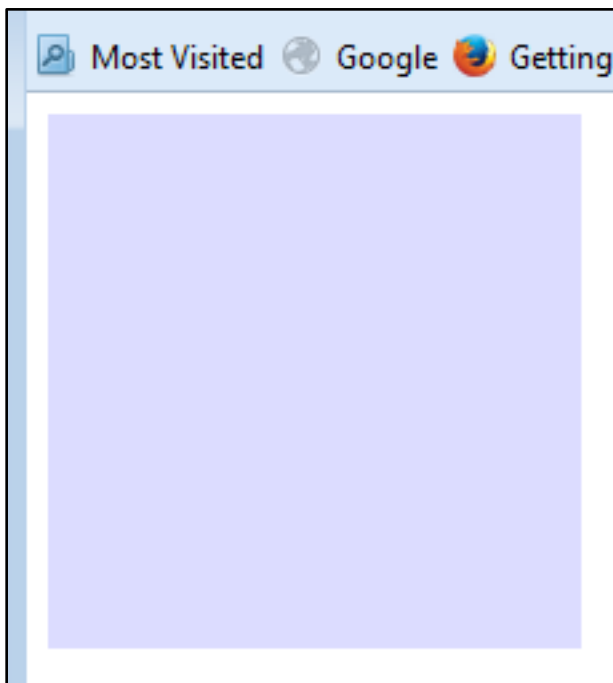
10) The JavaScript / Processing.js code below runs without errors and draws an image on a 200x200 pixel HTML canvas. Sketch a reasonable approximation of what is in the screen capture of the canvas below.

```
1)    background(220, 220, 255); //red, green, blue
2)    stroke(0,0,0);
3)    var x = 80;
4)    while (x <= 120)
5)    {
6)        var y = 0;
7)        while (y <= 180)
8)        {
9)            line(x, y,      x,    y + 10); //x1,y1,x2,y2
10)           line(x, y + 5, x-5, y + 10);
11)           line(x, y + 5, x+5, y + 10);
12)           y = y + 20;
13)       }
14)       x = x + 40;
15)   }
```