

Name: _____

With solutions

**CS110 Introduction to Computing
Fall 2011
Exam 1**

This is a closed book / closed notes exam. Your work on this exam must be your own. Answer all questions in the space provided, continuing your answer on the back of the page if necessary. If you want scrap paper, please ask your instructor.

All Processing statements you write should be syntactically correct. There will be no credit for incorrect use of Processing syntax.

Be certain to use proper indentation in any code you write. Comment your code wherever necessary to ensure proper understanding.

Be certain to sign the log-in sheet for the examination period, and the statement of academic honesty on the exam cover page.

I wish you each the best of luck!

Sign the following statement **after** you have completed the examination. Your exam will **not be graded** without your signature.

I certify that my responses in this examination are solely the product of my own work and that I have fully abided by the Bryn Mawr College Computer Science Academic integrity policy and instructions stated above while taking this exam.

Signature: _____

Printed Name: _____

For Instructor Use Only:

Question #	Points	Maximum Points
1.		10
2.		10
3.		15
4.		15
5.		10
6.		10
7.		15
8.		20
Total		100

Name: _____

Section 1: Basics (10 points total) Circle or write the correct answer to each question.

1.1. (2 pts) Write your name on EVERY page of the exam booklet and any scrap paper you use.

1.2. (2 pts) What would Processing do as a result of these commands? **B**

```
rectMode(CORNER);  
rect(0, height/2, width, height/2);
```

- A. Draw a rectangle that covers the top half of the window
- B. Draw a rectangle that covers the bottom half of the window
- C. Draw a rectangle that covers the left half of the window
- D. Draw a rectangle that covers the entire window

1.3. (2pts) What would the Processing window look like as a result of these commands? **D**

```
stroke(0);  
line(0, 0, width, height);  
background(255);
```

- A. A diagonal black line on a white background
- B. A vertical black line on a white background
- C. A horizontal black line on a white background
- D. A white background

1.4. (2 pts) How many times would this code print the text “Hello World”? **B**

```
for (int i=10; i > 0; i -= 3) {  
  println("Hello World");  
}
```

- A. Three (3)
- B. Four (4)
- C. Five (5)
- D. None, this is invalid Processing code.

1.5 (2 pts) What value is printed as a result of the following code? **B**

```
int z = 7 / 3;  
println(z);
```

- A. 7
- B. 2
- C. 2.5
- D. None, this code will result in an error.

Section 2: Operators and Expressions (10 points total)

What is the value of x after the following statements are evaluated?

2.1 (2 pts) `int x = 11;`
 `x++;`

12

2.2 (2 pts) `double x = 7;`
 `x -= 2;`

5

2.3 (2 pts) `int x = 13;`
 `x = x % 5;`

3

2.4 (2 pts) `double y = 7.5;`
 `boolean x = (y > 8);`

false

2.5 (2 pts) `int y = 0, z = 1;`
 `boolean x = (y > 0 || z < 0);`

false

Section 3: Conditionals (15 pts total)

3.1 (7 pts) Write a conditional expression that prints to the screen “positive even” whenever the integer variable *num* is even and greater than zero, prints “positive odd” when it is odd and greater than zero, and prints “non-positive” otherwise.

```
int num = ... ; // num has been initialized previously to some value

// write your code below here

// Outer if-statement tests for positive/non-positive
if (num > 0) {

    // Inner if-statement tests for even/odd
    if (num % 2 == 0) {
        println("positive even");
    } else {
        println("positive odd");
    }

} else {
    println("non-positive");
}
```

3.2 (8 pts) The following program was written by an artist to progressively form an abstract rendition of the Flag of Italy from colored circles. The circles are drawn and filled with appropriate colors as the mouse is moved around the sketch. Italy's flag is referred to in Italian as *il Tricolore* because it features three equally sized vertical pales of green, white, and red. Green fills the left third, red fills the right third, and white fills the center.

Is the artist also a competent programmer? Why or why not? Explain your answer.

The artist is not competent. The last of the three if-statements will always be true because mouseX is always > 0.0. Consequently, the ellipse fill color will always be set to green.

```
void setup() {
    size(500, 300);
    background(0);
}

void draw() {}

void mouseMoved() {
    noStroke();
    if (mouseX > 0.67 * width) {
        fill(255, 0, 0);
    }
    if (mouseX > 0.33 * width) {
        fill(255);
    }
    if (mouseX > 0.0){
        fill(0, 255, 0);
    }
    ellipse(mouseX, mouseY, 20, 20);
}
```



Section 4: Loops (15 pts total)

4.1 (10 pts) Rewrite the following code, using a while-loop instead of a for-loop. The results of executing the two code fragments should be identical.

```
int k = 30;
for(int i=10; i>0; i-=2){
    k -= i;
    println(k);
}
```

```
int k = 30;
int i = 10;

while (i>0) {
    k -= i;
    println(k);
    i-=2;
}
```

4.2 (5pts) What will be printed by the code fragment in the above question 4.1?

20
12
6
2
0

5. Functions (10 pts) Consider the following code. When this program runs, what will be printed?

```

int x = 1;                                0
                                           -5

void setup() {
    int x = -5;
    test(x);
    println(x);
}

void draw() { }

void test(int x) {
    x += 5;
    println(x);
}

```

6. Arrays and Random Numbers (10 pts) Complete the following function.

```

/** Returns an array of n random numbers between 0 and 1
 * @param n - the quantity of random numbers to generate
 * @return an array of n numbers, each of which is between
 *          0 and 1 and was generated randomly
 */
float[] generateRandomNumbers(int n) {

    // Your code goes here. Be sure to comment your code!

    // Create a new float array of size n
    float[] tmp = new float[n];

    // Fill array with randomly generated numbers in [0,1]
    for (int i=0; i<n; i++) {
        tmp[i] = random(1);
    }

    // Return the new array
    return tmp;

} // closing brace for generateRandomNumbers()

```

7: Random Numbers and Debugging (15 pts)

The following program is *supposed* to draw stars across a black sky at random locations with a green ground beneath it. But it doesn't work correctly, and only draws the sky and ground.

```
int numStars = 500;
float maxStarDiameter = 5; 0.5;

void setup() {
  size(500, 500);

  // draw the black sky and grass
  background(0);
  fill(0,170,0);
  rectMode(CORNER);
  rect(0, height/2, width, height/2);

  // draw the stars as small circles
  for (int i = 0; i < numStars; i++) {

    // choose the star's location size and color
    int x = (int) random(width);
    int y = (int) random(height/2);
    int diameter = (int) random(maxStarDiameter);
    int starColor = (int) random(200, 255); 0,1;

    // draw the star
    fill(starColor);
    ellipse(x, y, diameter, diameter);
  }
}

void draw() {}
```

7.1 (8 pts) There are three logical errors in this program that prevent the stars from being drawn correctly. What are they? Be certain to explain each error.

- 1. The value of maxStarDiameter is too small for the random() function. It produces values for diameter values for the ellipse that are also too small.**
- 2. The parameters passed to the random() function for computing starColor are too small. Star colors should be near white, which is specified by a color value near 255.**
- 3. The random() function used to compute y values for star locations can generate values that cause stars to be drawn on the ground. The random() function range for computing values for y should be limited to the range (0, height/2);**

7.2 (7 pts) In the code above, correct the three errors. Justify your answers if you're not certain.

8: Classes (20 pts)

Write a class called Balloon that has three integer fields named x, y, and d that represent the Balloon's location (x, y), and its diameter (d). The Balloon constructor should initialize d to a value of 1, and x and y to values passed to the constructor as two arguments.

The Balloon class has one method named display() that draws the Balloon as an ellipse at its location (x, y) with a diameter of d. The display() method also increments the d field by 1 every time it is called. When a Balloon object's d field exceeds a value of 100, d is reset to 1.

A start of the class has already been defined for you, along with a program to use the Balloon class. You may use the back of the page if necessary.

```

Balloon myBalloon;

void setup() {
    size(500, 500);
    smooth();
    myBalloon = new Balloon(random(width), random(height));
}

void draw() {
    background(200);
    myBalloon.display();
}

class Balloon {
    // Your implementation goes here

    int x;                // Declare fields
    int y;
    int d;

                                // Constructor
    Balloon(float tx, float ty) {
        x = (int)tx;
        y = (int)ty;
        d = 1;
    }

    void display() {          // Display Balloon
        fill(200, 255, 200);
        ellipse(x, y, d, d);

        // Increase diameter of Balloon
        // When 100 is reached, reset to 1
        d++;
        if (d >100) {
            d = 1;
        }
    }
}

} // closing brace for Balloon class

```