

Processing functions reference

ellipse(a, b, c, d)

Parameters

- a** float: x-coordinate of the ellipse center
- b** float: y-coordinate of the ellipse center
- c** float: width of the ellipse
- d** float: height of the ellipse

rect(a, b, c, d)

Parameters

- a** float: x-coordinate of the rectangle top left corner
- b** float: y-coordinate of the rectangle top left corner
- c** float: width of the rectangle
- d** float: height of the rectangle

0.) THE EASY ONE (1 point)

- Check that your exam has all x pages (excluding the cover sheet and scratch paper).
- Write your name on the front of the exam.
- Sign the certification that you comply with the Penn Academic Integrity Code.

1.) MISCELLANEOUS (8 points total)

1.1) (2 points) What would be the result of the following code fragment in a Processing `setup()` function?

```
size(500, 500);  
fill(0, 0, 255);  
rect(0, 0, 500, 250);
```

- A. a window with the top half blue and the bottom half white**
- B. a window with the bottom half blue and the top half white
- C. a window that is entirely blue
- D. a window that is entirely white

1.2) (2 points) What would the window look like as a result of the following code fragment in a Processing `setup()` function?

```
size(500, 500);  
line(200, 250, 300, 250);  
line(250, 200, 250, 300);  
ellipse(250, 250, 50, 50);
```

- A. A circle divided into 4 quarters**
- B. A circle divided into 2 halves
- C. Just a circle. The lines will not be visible
- D. Two intersecting lines. No circle

1.3) (2 points) How many circles are drawn as a result of this code fragment in a Processing `setup()` function?

```
for (int i = 5; i < 40; i *= 2) {  
    ellipse(i, i, 10, 10);  
}
```

- A. 7
- B. 3**
- C. 4
- D. 8

1.4) (1 point) The result of evaluating $5 * 3 + 2 / 3$ is _____15_____.

2.) OPERATORS AND EXPRESSIONS (10 points total)

For each code fragment in the left column, (a) fill in the most appropriate data type in the middle column and (b) give the value that z contains after the code has been executed in the right column.

If the code would result in an error, write “ERROR” in the middle column and give the reason for the error in the right column (you do not need to write the exact error message). The first two problems have been completed for you.

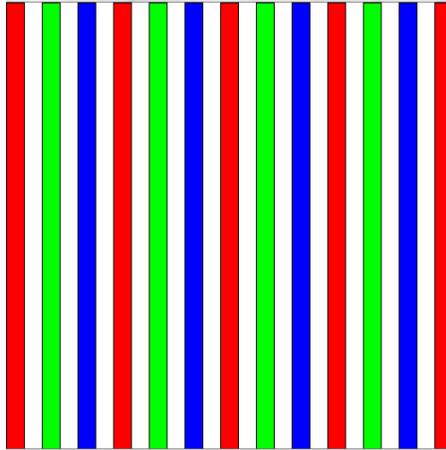
<pre> _____ z = 11; z++; </pre>	int	12
<pre> int x = 100; _____ z = x.length; </pre>	ERROR	x doesn't have an x.length field
<pre> _____ w = 'a'; </pre>	char	N/A
<pre> boolean b = true; _____ z = b && (4 < 5); </pre>	boolean	true
<pre> String s = " = Perfect!"; s = 4 + 2 + s; _____ z = s; </pre>	String	6 = Perfect!
<pre> int x = 5; int y = 2; _____ z = x / y * 3.0; </pre>	double	6.0
<pre> int[] a = {1, 2, -1, 0, 5}; _____ z = a.length + a[1]; </pre>	int	7
<pre> int i = 5; int z = 6; z += i + 1; </pre>	N/A	12

3.) CONDITIONALS, LOOPS, AND ARRAYS (14 points total)**3.1) Stripes (8 points)**

We want to draw the following colorful striped image in Processing. Each stripe is to be 20 pixels wide. Complete the `setup()` method below in order to achieve this. Remember that red is (255,0,0), green is (0,255,0) and blue is (0,0,255).

Do not write 40 lines of code that draw rectangles. Elegant code will be rewarded!

You may continue your code on the next page, which is blank.



```
public void setup() {
    int width = 500;
    int height = 500;
    size(width, height);
    background(255);

    //There is a 120 pixel gap between stripes of the same primary colour.
    //red stripes
    fill(255, 0, 0);
    for (int x = 0; x < width; x += 120)
        rect(x, 0, 20, height);
    }
    //green stripes begin at x = 40
    fill(0,255,0);
    for (int x = 40; x < width; x += 120){
        rect(x, 0, 20, height);
    }
    //blue stripes begin at x = 80
    fill(0,0,255);
    for (int x = 80; x < width; x += 120){
        rect(x, 0, 20, height);
    }
}
```


3.2) (10 points)

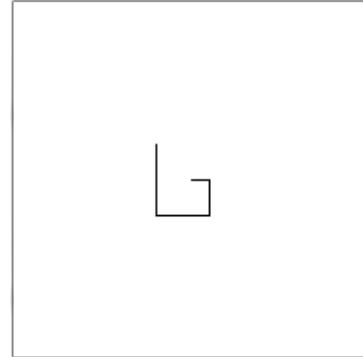
In Processing, we are given two arrays as class-level variables. One of them, `char[] dir`, has characters for directions. The other, `int[] steps`, gives the number of pixels to draw in that direction. We want to draw a path, beginning from the center of the window.

For instance, for these specific values

```
char[] dir = {'r', 'd', 'l', 'u'}; //directions
int[] steps = {10, 20, 30, 40}; // steps taken in pixels
```

we would like a picture that looks like the image on the right, where from the center we draw 10 pixels to the right, then 20 pixels down, then 30 to the left, then 40 up.

Complete the `setup()` function below to achieve this.



```
public void setup() {
  float x = 100;
  float y = 100;
  size(200, 200);
  for (int i = 0; i < dir.length; i++){
    switch(dir[i]){
      case 'l':
        line(x, y, x - steps[i], y);
        x -= steps[i];
        break;
      case 'r':
        line(x, y, x + steps[i], y);
        x += steps[i];
        break;
      case 'u':
        line(x, y, x, y - steps[i]);
        y -= steps[i];
        break;
      case 'd':
        line(x, y, x, y + steps[i]);
        y += steps[i];
        break;
    }
  }
}
```

4.) TRACERY (9 points)

Trace through the following code. Assume that the program is executed using the command:

```
java Tracery 7 8 9
```

In the table to the right of the code, write the values of a, b, and c in the order that they would appear on the console (Interactions pane of DrJava) at each of the `System.out.println()` statements. You may not need to use all of the table's rows.

```
public class Tracery {
    static int a = 0;
    static String b = "";
    static String c = "";

    static int f1(boolean flag, String c) {
        if (!flag) {
            b = "" + a;
            a = f2(a, b);
        } else {
            a = f2(a, c);
        }
        System.out.println(a + " " + b + " " + c);
        return a;
    }

    static int f2(int b, String s) {
        c = s;
        System.out.println(a + " " + b + " " + c);
        return Integer.parseInt(s) + b;
    }

    public static void main(String[] args) {
        a = Integer.parseInt(args[0]);
        b = args[1];
        c = args[2];
        a = f1(true, b);
        System.out.println(a + " " + b + " " + c);
        a = f1(false, b);
        System.out.println(a + " " + b + " " + c);
    }
}
```

a	b	c
7	7	8
15	8	8
15	8	8
15	15	15
30	15	8
30	15	15

5.) DEBUGGING (10 points)

Find and correct the errors in the following program that cause behavior different to that described in its header comment. Some of the errors can be corrected by adding, removing, or rewriting a line of code or a symbol such as a semicolon. Write `//` before any line of code that should be removed. Write any inserted or rewritten code in the space between the lines.

If you require more space, you may fill in the blanks on the next page.

There are a total of 8 issues in this program.

```

/*
 * This program takes a single command line argument N and simulates N
 * coin tosses by using random numbers.
 * Each time the coin shows heads you earn a dollar.
 * Each time the coin shows tails you lose a dollar.
 * After N coin tosses, the total amount you earned is printed.
 */

1  public class Gambling {

2      public static void main(String[] args) {

3          int numOfArguments = args.len

4          float totalAmount = 0;

5          if numOfArguments > 1 System.out.println("only 1 argument please");

6          else totalAmount = tossCoins(args);

7          System.out.println("amount earned is " + totalAmount);

8      }

9      public static int tossCoins( n ) {

10         int winnings = 0 ;

11         for (int i =1; i <= n; i++) {

12             double randomNumber = math.random();

13             if (randomNumber > 0.5) winnings += 1;

14             else winnings -= 1;

15         }

16         return winnings;

```

15 }

16 }

Line 3 – `args.length;`

Line 5 – `if (numOfArguments > 1)`

Line 6 – `tossCoins(args[0])`

Line 9 - `int n`

Line 11 – `Math.random();`

Winnings has to be declared as a variable at the start of the function body and then returned at the end (as shown above)

6.) FUNCTIONS! (20 points total)

6.1) (8 points) Write the public static function that is described by the function header below, including the function declaration. You do not need to define any class.

```
/** Function Name:  occurrences
 * Parameters:
 *   x - an integer
 *   arr - an array of doubles
 * Returns:
 *   the number of elements of arr that are within 0.5 of x.
 * occurrences(3, {2.5, 3.2, 1, 0.0}) -> 2
 * occurrences(-1, {0, 0.5, -1.5}) -> 1
 */
```

```
public static int occurrences(int x, double[] arr){
    int count = 0;
    for(int i = 0; i < arr.length; i++){
        double dist = x - arr[i];
        if ( Math.abs(dist) <= 0.5) {
            count++;
        }
    }
    return count;
}
```

6.2) (12 points) Write the public static function that is described by the function header below, including the function declaration. You do not need to define any class.

```
/** Function Name:  shiftSentence
 * Parameters:
 *   strArr - an array of strings of length n
 *   begin - a positive integer less than n
 * Returns:
 *   a shifted sentence using strings in strArr as shown in examples
 * Error checking:
 *   if begin is greater than or equal to n, return ""
 * Examples:
 * shiftSentence({"exams", "are", "tough"}, 2)  returns
 * "tough exams are"
 * shiftSentence({"a", "b", "c"}, 0)  returns
 * "a b c"
 * shiftSentence({"Hurrah", "for", "the", "red", "and", "the",
 * "blue"}, 3) returns
 * "red and the blue Hurrah for the"
```

```
*  shiftSentence({"I", "find" , "your", "lack", "of", "faith",  
"disturbing"}, 2) returns  
  "your lack of faith disturbing I find"  
*/  
  
public static String shiftSentence(String[] strArr, int begin){  
    if (begin >= strArr.length){  
        return "";  

```

Scrap Paper

(This page is intentionally blank)