Lab 14: Functions 3

Consult the CS 105 LEARN Site for lab due dates and standard lab instructions.

1. **Exercises in simple sketches.**
   Unless stated otherwise, each sketch uses a 100 x 100 canvas and background 220. Don't forget to use good style (e.g. indent, whitespace, semicolons, let).

   **SAVE each sketch as “lab_14_1a”, “lab_14_1b”, etc.**

   a. Define a function that draws a snowflake with its centre at a position defined by two parameters, x and y. The snowflake is created using two 8 pixel long lines and a 4 pixel diameter circular centre. Draw the snowflake with stroke and fill semi-transparent white.

   Write a setup function that uses your snowflake function in a loop to draw 100 randomly placed snowflakes around the canvas. Make the background “midnight blue” (use #0b28ba).
b. Define a function called `arrow` that draws a 50 by 50 arrow in the centre of a 100 by 100 canvas, so that it points either North, South, East, or West depending on a single parameter. If the function is called with "N" as the argument, it points North, if "E" it points East, and so on. Your submission should have the code provided below and your `arrow` function.

HINT: Use graph paper to figure out how to draw the four different arrow directions.

HINT: In your function, you can test if the direction parameter is equal to "N", "S", "E", or "W" using the equality relational operator.

Test the four different ways to call your function by changing the code below:

```javascript
function setup() {
  createCanvas(100, 100);
  background(220);
  arrow("N"); // or "S", "E", "W"
}
```

```
arrow("N")  arrow("S")  arrow("E")  arrow("W")
```

c. Define a function called `coinFlip()` that returns a random boolean value with equal probability (half the time it returns true, half the time it returns false).

Write a setup function that uses your `coinFlip()` function 150 times in a loop. Keep track of how many times the value returned was heads and how many times it was tails. Draw your results as a simple horizontal bar graph with heads at the top and tails at the bottom, labelling each bar as below. **Do this all in the setup function. You do not need to use the draw function.** If done correctly, each time the program runs, the number of heads and tails will differ, as shown below.

HINT: Use `textAlign` to make it easier to **vertically CENTER** your text in each bar.
d. Define a function called donutHitTest that returns true if the mouse is on the donut, not including the hole (see the images below). The function takes 6 parameters: the x and y position for the point to test, the x and y position of the center of the donut, and the outer and inner radius of the donut.

Your function should work with the code below, so the donut turns pink when the mouse is on the donut, not including the hole. You should not change setup or draw. Your submission should have the code below and the donutHitTest function you have defined. When we run the program, it should behave as shown by the images below.

```
function setup() {
  createCanvas(100, 100);
}

// centre of donut
let x = 50;
let y = 50;

function draw() {
  background(220);

  if (donutHitTest(mouseX, mouseY, x, y, 70, 30)) {
    fill("#ff8ccd"); // pink
  } else {
    fill(255); // white
  }

  // draw the donut
  ellipse(x, y, 70, 70); // outer
  fill(220);
  ellipse(x, y, 30, 30); // inner hole
}
```

![donut images](image1.png) ![donut images](image2.png) ![donut images](image3.png)
e. Write code to perform a hit test on the irregular shape drawn by the code below. When the mouse is inside the shape, change the fill to fuschia (use #d200fc). Your submission should contain the code below, plus your code, and behave as shown by the images below.

HINT: Use multiple calls to the `circleHitTest` function from the lecture.

```javascript
function setup() {
  createCanvas(100, 100);
  noStroke();
}

function draw() {
  background(220);

  // do the hit testing right here in draw
  fill(0); // black

  // draw the irregular shape
  ellipse(50, 60, 45, 45);
  ellipse(70, 40, 30, 30);
  ellipse(30, 40, 30, 30);
}
```

![Images showing the irregular shape with different fill colors.](image1.png)

f. Write a program to randomly draw exactly 500 points inside a circular area with a radius defined by a variable called `r`.

HINT: Call random to pick a possible x and y coordinate anywhere in a square area that fits around the circle, then add if that position is inside the circular area. Use the `circleHitTest` function from the lecture.

```javascript
r = 40
r = 25
r = 10
```

![Images showing points distributed at different radii.](image2.png)
Submitting

Submit a single ZIP file called “lab_14.zip” containing all your sketches, see “How to Save and Submit” (on CS 105 LEARN Site).

It is your responsibility to submit to the correct files to the correct dropbox before the deadline. Otherwise, you will receive a mark of 0.