Name:_**Solution**________________________________

CS110 Introduction to Computing
Fall 2016
Exam 1

This exam is closed book/notes. No computers/devices are permitted.. Answer all questions in the space provided, continuing on the back of the pages if necessary.

This exam contains 9 questions (pages 1-11) to be taken in 80 minutes.

Assume that any given Processing statements do not have typographical errors. If you cannot complete all of a problem, do as much as you can. You may receive partial credit.

Good Luck!

Question #	**Points**	**Maximum Points**
1		**10**
2		**5**
3		**5**
4		**10**
5		**15**
6		**15**
7		**10**
8		**15**
9		**15**
Total		**100**

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 Academic Integrity policy and instructions stated above while taking this exam.
>
> Signature:________________________________
>
> Printed Name:_____________________________

Question 1 (10 points) Write the correct answer in each of the following:

a) Write a Processing command to create a canvas of 300x300 pixels:

```
size(300,300);
```

b) After the above command is carried out, what will be the value of the variable, **`width`**:

____**300**__________________________

c) Write the command to translate the origin to the coordinates <135, 45>:

```
translate(135,45);
```

d) What will be printed by the following code segment:

```
int i = 1;
while (i <= 5)
{
  i++;
}
println(i);
```

6

e) Consider the commands below:

```
int i=1;
while (i < 15) {
   if (i%3 == 0) {
      println("Aloha");
   }
   i = i + 1;
}
```

How many times would this code print the text "`Aloha`"? ______**4**______

Question 1 (*cont'd)*
Suppose the function defined as:

```
void spongeBob(float x, float y, float s) {
     …details deleted…
} // spongeBob()
```

draws the Sponge Bob cartoon character with its top-left corner at <`x`, `y`> and of size `s` (i.e., its width and height are both `s`). Write the commands to do the following:

In the function definition above:

f) What is the name of the function defined above? ____**`spongeBob`**_______

g) How many parameters does `spongeBob()` take? _____3_______________

h) Use the function to draw a Sponge Bob of size 125x125 pixels with its top-left corner at <100, 90>:

```
spongeBob(100,90,125);
```

i) Use the function to draw a Sponge Bob of size 175x175 pixels with its top-left corner at <100, 300>:

```
spongeBob(100,300,175);
```

j) Use the function to draw an *upside-down* Sponge Bob of size 100x100 pixels with its *bottom-right* corner at <200, 240>.

```
rotate(PI);
spongeBob(-200,-240,100);

// OR

translate(200,240);
rotate(PI);
spongeBob(0,0,100);
```

Question 2 (5 points)
Write a conditional expression that would print "`Just Right`" whenever an integer variable `temperature` has a value between 67 and 74, inclusive.

```
     int temperature = …;
     // write your conditional here:
```

```
if(temperature >= 67 && temperature <= 74)
{
     println("Just Right");
}

// OR

if(temperature >= 67)
{
  if(temperature <= 74)
  {
    println("Just Right");
  }
}
```

Question 3 (5 points)
Rewrite the following using a while-loop:

```
for (int i=1; i <= 18; i++) {
   if (i%3 == 0) {
      println("Hello");
   }
}
```

```
int i = 1;
while(i <= 18)
{
     if(i % 3 == 0)
     {
          println("Hello");
     }
     i++;
}
```

Question 4 (10 points)

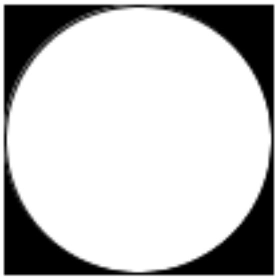
Write Processing code segment to draw a line from the center of a sketch to its border. It should be drawn tilted by a given angle, `theta`, specified in radians. You may assume that the variable `theta` is already defined. Your code should work for any size sketch window, and it is acceptable for the line to pass the border, but it must reach at least to the sketch border.

```
translate(width/2,height/2);
rotate(theta);
line(0,0,width+height,0);
  // the width + height part is to ensure the line is
  // long enough

// OR

translate(width/2,height/2);
line(0,0,(width+height)*cos(theta),
         (width+height)*sin(theta));
```

Question 5 (15 points)
The Processing program below is supposed to draw the following figure centered at every mouse click:



The width and height of the figure should always equal each other, and it should be a random number chosen between `50` and `150`.

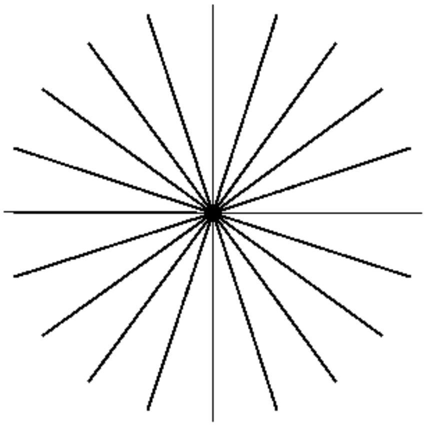
Complete the program below by writing the appropriate commands needed to accomplish the task. Continue on the back of page if more space is needed.

```
void setup() {
   size(400, 400);
   background(255);
} // setup()

void draw() { }

void mousePressed()
{
     float size = random(50, 150);
     translate(mouseX, mouseY);
     fill(0);
     rectMode(CENTER);
     rect(0, 0, size, size);
     fill(255);
     ellipse(0, 0, size, size);
}
```

Question 6 (15 points)
Write a complete Processing function that draws the following figure:



The function should be defined as shown below. The sketch shown was generated using the call:

```
drawSpokes(250, 250, 200, 20);

void drawSpokes(int x, int y, int s, int n) {
   // draws n spokes centered at <x, y> of length s/2

   translate(x, y);
   for(int i = 0; i < n; i++)
   {
      line(0, 0, s/2, 0);
      rotate(2 * PI / n);
   }















} // drawSpokes()
```

Question 7 (10 points)

Consider the following code segment. Using the table below, trace the code segment to determine <u>what will be printed</u> by the call to `mystery(3,8)` :

```
void mystery(int a, int b)
{
     int x = a;
     for(int i = 0; i < b - a; i++)
     {
          println(x);
          if(i % 2 == 0)
          {
               x = b - i;
          }
          else
          {
               x = a + i;
          }
     }
     println(x);
} // mystery()
```

Write what is printed here:

```
3
8
4
6
6
4
```

a	b	x	i
3	8	3	0
		8	1
		4	2
		6	3
		6	4
		4	5

Question 8 (15 points)

Write a Processing function `sum(int n)` that computes and prints the summation:

$$1^2 + 3^2 + 5^2 + \cdots + n^2$$

```
void sum(int n)
{
     int s = 0;
     for(int i = 1; i <= n; i += 2)
     {
          s += i * i;
     }
     println(s);
}
```

Question 9 (15 points)
Write a Processing <u>function</u> `isPrime(int n)` that detects whether a number `n` is prime. A prime number has no even divisors except for 1 and the number itself. When calling `isPrime()` and passing in a prime parameter, `isPrime()` should print "`prime`". Otherwise, it should print "`composite`". So, `isPrime(7)` prints "`prime`" and `isPrime(9)` prints "`composite`". Do not worry about what happens if the parameter is less than 2.

```
To see if a number N is prime do the following
  for each number (call it i) starting at 2 and counting up to (N-1)
        if i divides N evenly then N is not a prime number
  otherwise, N is prime
```

```
void isPrime(int n)
{
      boolean prime = true;
      for(int i = 2; i < n; i++)
      {
            if(n % i == 0)
            {
                  prime = false;
            }
      }

      if(prime)
      {
            println("prime");
      }
      else
      {
            println("composite");
      }

      // OR

      for(int i = 2; i < n; i++)
      {
            if(n % i == 0)
            {
                  println("composite");
                  i = n; // leave loop
            }
      }
      if(i == n) // this means we left "naturally"
      {
            println("prime");
      }


} // isPrime()
```

PROCESSING QUICK REFERENCE

Variables/Constants:

```
width, height, mouseX, mouseY, HALF_PI, PI, QUARTER_PI,
TWO_PI
```

Data Types

```
boolean, int, float
```

Conversion Functions

```
int(x), float(x), double(x), degrees(rad), radians(deg)
```

Relational and Logical Operators

!= (inequality)	< (less than)	<= (less than or equal to),
== (equality)	> (greater than)	>= (greater than or equal to)
! (logical NOT)	&& (logical AND)	|| (logical OR)

Loops

```
for (init; test; update) {         while (expression) {
  statements                           statements
}                                   }
```

Conditionals

```
if (test) {            if (expression) {       if (expression) {
  statements             statements              statements
}                      }                       }
                       else {                  else if (expression){
                         statements              statements
                       }                        }
                                                …
                                                else {
                                                  statements
                                                }
```

2D Primitives

```
arc(x,y,width,height,startRad,endRad)
ellipse(x,y,width,height)
line(x1,y1,x2,y2)
point(x,y)
quad(x1,y1,x2,y2,x3,y3,x4,y4)
rect(x,y,width,height)
triangle(x1,y1,x2,y2,x3,y3)
text(message,x,y)
```

PROCESSING QUICK REFERENCE

Attributes
`ellipseMode(mode), rectMode(mode)`

Vertex
`beginShape(), endShape(), vertex(x,y), curveVertex(x,y)`

Mouse
```
mouseClicked(), mouseDragged(), mousePressed(),
mouseReleased(), mouseX, mouseY
```

Text Output
`print(message), println(message)`

Setting
```
background(r,g,b), fill(r,g,b), noFill(), noStroke(),
stroke(r,g,b)
```

Trigonometry
`cos(rad), sin(rad), tan(rad)`

Random
`random(lower,upper), noise(t)`