CS110 – Fall 2015
Problem Set 4 (For Practice)

Name: _____

1. Declare all the necessary classes in order to make the following driver program work properly (steps have been broken down for you in the sub-parts). For example, this is a sample output that is acceptable:

Generally, a Dolphin can be found in water, it can not lay eggs, and is often overheard saying 'ak, ak, ak, ak'
Generally, a Platypus can be found on land, it can lay eggs, and is often overheard saying 'errrr'
Generally, a Human can be found on land, it can not lay eggs, and is often overheard saying 'I'll take a grande latte with a double-shot of espresso'
Generally, a CSStudent can be found on land, it can not lay eggs, and is often overheard saying 'I love programming!'

```
void setup() {
  Mammal[] mammals = new Mammal[4];

  mammals[0] = new Dolphin();
  mammals[1] = new Platypus();
  mammals[2] = new Human();
  mammals[3] = new CSStudent();

  for (int i=0; i< mammals.length; i++) {

    print("Generally, a " + mammals[i].getName() );
    print(" can be found ");

    if( mammals[i].livesInWater() == false ) {
      print("on land, ");
    } else {
      print("in water, ");
    }

    print("it can ");
    if( mammals[i].laysEggs() == false ) {
      print("not ");
    }
    print("lay eggs, and is often overheard saying '");
    mammals[i].speak();
    println("'");

  }
}
```

1.1 (20 pts) Declare a new class called Mammal with the following members:

- Two String fields called 'name' and 'sound'
- A constructor that accepts two String parameters ('name' and 'sound') and saves values in fields
- A void method called 'speak()' that prints the object's sound to the console area,
- A boolean method called 'laysEggs()' that returns false
- A boolean method called 'livesInWater()' that returns false.
- A 'getter' String method called 'getName()' that returns the object's name field;

1.2 (10 pts) Declare a new class called Platypus that extends Mammal. Override methods as appropriate.

1.3 (10 pts) Declare a new class called Dolphin that extends Mammal. Override methods as appropriate.

1.4 (10 pts) Declare a new class called Human that extends Mammal. Override methods as appropriate.

1.5 (10 pts) Declare a new class called CSStudent that extends Human. Override methods as appropriate.

2. Rewrite the following function:

```
void drawSpokes(int centerx, int centery, int radius) {
  int steps = 30;
  int size = radius*2;
  float angle = 2*PI/steps;
  stroke(random(255), random(255), random(255), 127);
  for (int i=0; i<steps; i++) {
    float x = sin(angle*i)*radius + centerx;
    float y = cos(angle*i)*radius + centery;

    line(centerx, centery, x, y);
  }
}
```

2.1 (20 pts) Define a corresponding Spokes class . The class should contain the following data fields: x, y, radius, steps, angle and c (for color), of which x, y, radius and steps should be set via the constructor.

2.2 (25 pts) Write its draw() function with transformations instead of coordinate calculation with sins and cosines.

2.3 (15 pts) Add a setup() to complete a program with will create 50 spokes of random location in the sketch window, with size varying from 20 to 50 and number of spokes varying from 10 to 30. Store them in an array and draw.

3. (40 pts) Write a recursive function int gcd(int x, iny), which returns the greatest common divisor of x and y.

4. (40 pts) Complete the following program by adding four recursive calls to the end of the drawBoxes() function. Each recursive drawBoxes() call should be centered at the current box's four corners, and it should draw boxes half the size of the original. The output should look like the image provided.

```
void setup() {
  size(500, 500);
  background(255);
  rectMode(CENTER);
  noFill();
  stroke(0);

  // Kick off the recursive draw
  drawBoxes( 0.5*width, 0.5*height, 0.5*width );
}

// Draw boxes recursively,
// centered at cx, cy, with width/height of d.
void drawBoxes(float cx, float cy, float d) {
  strokeWeight(0.1*d);
  stroke(d);
  rect(cx, cy, d, d);

  // Base case.
  if (d < 20) return;

  // Add four recursive calls to drawBoxes() that repeats
  // drawing with boxes half the size of the current box,
  // centered at the current box's four corners.

}
```