Your Name: ____________________

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Georgia Tech community.

CS 1316 - Exam 1 - Fall 2009

Problem	Earned Points	Possible Points
1. Vocabulary		15
2. Fill in the Blank		6
3. Code Tracing		5
4. Turtle Graphics		6
5. Draw Line		15
6. My Counter		10
7. Identify Syntax Problems		8
8. Build a Subclass		10
Total:		**75**

1. Vocabulary (15 points)

For each of the following words, write a 1-2 sentence definition of the word as used in this class. Your definition should be concise and to the point, while demonstrating that you know what the term means.

a) method signature - The combination of method name, return type, and parameter types and ordering that uniquely identifies a particular method.

b) static - A keyword which indicates that a field or method belongs to a class, and not to objects instantiated from the class.

c) model - A detailed description of structure and behavior. (i.e. A representation of physical objects or systems that is used (via a simulation, or executing the model) to answer questions about the real objects or systems.)

d) field - A variable that is inside of a class, a.k.a. an object variable.

e) object - An object is an instance of a class, and has it's own data (object variables, or fields) and behavior (object functions, or methods).

Grading: 3 points for a very good definition. 2 points if you think they know what the term means, even if the definition isn't very good. 1 point if they get some of the right keywords in the definition. 0 points for completely wrong definitions or blanks.

2. Fill in the Blank (6 points)

I am in Section A__<correct section>_ and my grading TA's name is: _<name>_.
In Java, a semicolon (;) character is used to ___ end a statement _______.
The operator && indicates a _logical AND_, while the || operator indicates a _logical OR_.
Assume that an Employee class is a subclass of the Person class. A variable that is of type _Person___ can point to an object of type Employee and to an object of type Person.
Grading: 1 point per correct answer.

3. Code Tracing (5 points)

Trace through the following Java code and write down what it **prints** out.

```
int [] a = new int[10];

for ( int i = 0; i < 10; i++) {
     a[i] =  i;
}

for( int i = 0; i < 10; i++) {
     int index = a[9-i];
     a[i] = a[  index ];
}

for(int i = 0; i < 10; i++) {
   System.out.println( a[i] );
}
```

```
9
8
7
6
5
5
6
7
8
9
```

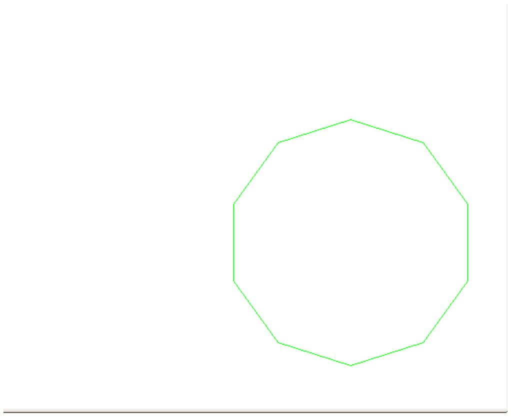
Grading: 5 points if correct. 4 points if numbers correct, but horizontal. 1 point if they get 9,8,7,6,5,4,3,2,1 or 1,2,3,4,5,6,7,8,9

4. Turtle Graphics (6 points)

The following code creates a turtle and uses it to draw a graphic. Assume the box on the right represents the World object given to the turtle. (Remember that turtles start in the center of their world, facing north/up). Draw the graphic drawn by the turtle. You do not need to draw the turtle.

```
Turtle t = new Turtle(new World());
for( int i = 0; i < 20; i++) {
  if (i %2 == 0) {
     t.turn(36);
  } else {
     t.forward(90);
  } // end if/else
} // end for
```

Grading:
+6 points for a decagon (ten line segments, 36 degree turns) that "starts" in the center, and is centered in the lower left quadrant.
+5 points for a decagon anywhere in the box.
+3 points for 10 line segments that don't close but that do all turn right).



+3 points if they are off by one line segment, but do close the polygon +2 points for 10 line segments that don't close and turn left.

5. Draw Line (15 points)

The variable **pic** already points at a Picture object that is some unknown number of pixels wide and some unknown number of pixels high. Write Java code that will draw a 5 pixel wide vertical red line directly down the middle of pic. (E.g. If the width of pic was 100, the line should be centered on the pixels with horizontal index 50, extending from pixels 48-52). You do not have to put the code inside of a function or class, just give us the code. You may assume the picture is at least 5 pixels wide and at least 1 pixel high.

```
int width = pic.getWidth();
int height = pic.getHeight();
int middle = width / 2;

for(int y = 0; y < height; y++) {
  for(int x = middle-2; x < middle+3; x++) {
    Pixel myPix = pic.getPixel(x,y);
    myPix.setRed(255);
    myPix.setGreen(0);
    myPix.setBlue(0);
  } // end for X

} // end for Y
```

Grading:
+2 - Calculating the middle as width / 2 (integer)
+2 - Starting x at middle -2
+2 - ending x at middle + 2
+2 - correctly iterating through all Y's
+2 - Getting each pixel out of the picture.
+3 - Setting red to 255
+2 - setting Blue & Green to 0

6. My Counter (10 Points)

Write a Java class called "Counter" that keeps track of how many objects of its type have been instantiated. To do this, create a default constructor for the class that adds one to a static integer class variable called "objectCount". (Make sure that "objectCount" is initialized with the value zero.) Also, write a static method called "getObjectCount" that returns your global variable as an integer.
After creating your Counter object, the following would be possible in the Interactions Tab in Dr. Java:

```
> Counter.getObjectCount()
0
> Counter myCt = new Counter()
> Counter.getObjectCount()
1
> Counter myCt2 = new Counter()
> Counter.getObjectCount()
2
```

```
public class Counter {
  public static int objectCount = 0;

  public Counter() {
    objectCount++;
  }

  public static int getObjectCount () {
    return objectCount;
  }
}
```

Grading:
+1 - class header correct
+3 - public static int objectCount = 0; (+2 if not explicitly initialized to zero)
+3 - constructor increments objectCount
+2 - public static int getObjectcount header correct.
+1 - getObjectCount returns objectCount.

7. Identify Syntax Problems (8 points)

A student wrote the following code for one of the class projects. He couldn't get it to compile, so he asked you for help. Please identify and fix all the errors.

```
public Book {

  private page_number;
  private int[] bookmarks;

  public Book(int page_number){
    page_number = page_number;
  }

  public addBookMarks(marks){
    bookmarks = marks;
  }

  public static void main()
  {
    Book book = new Book(150.0);

    int[] bookmark = new int[3]
    bookmark[0] = 20;
    bookmark[1] = 30;
    bookmark[2] = 50;

    book.addBookMarks(int[] bookmark);
  }

} // end class Book
```

```
public class Book {


  private int page_number;

  private int[] bookmarks;


  public Book(int page_number){

    this.page_number = page_number;

  }


  public void addBookMarks(int[] marks){

    bookmarks = marks;

  }


  public static void main(String[] args)
  {

    Book book = new Book(150);


    int[] bookmark = new int[3];

    bookmark[0] = 20;

    bookmark[1] = 30;

    bookmark[2] = 50;


    book.addBookMarks(<del> bookmark);

  }

} // end class Book
```

Grading: +1 point for each identified syntax or type error.

8. Build a Subclass (10 points)

Examine the Animal class in the appendix. Create a Dog class that is a subclass of Animal. It should have a constructor that accepts both a String name and an int age. It should also implement a "greet" method, which should say "Hi, I'm a dog named <name> and I'm <age> years old".

```
public class Dog extends Animal {
   public int age;

  public Dog(String n, int a) {
    super(n)
    age = a;
  } // end Constructor

  public void greet() {
    System.out.println("I'm a dog named " + getName() + " and I'm " + age
      + " years old!");
  } // end main

} // end class Dog
```

Grading:

+2 points: Class header is correct.
+1 points: class has an int age variable (public/private fine, but not static!)
+2 points: Constructor accepts a string and an int.
  +1 point: super() is called with string,
  +1 point: age is set with int.
+1 points: public void greet()method is declared.
  +2 points: getName() is used to get name from superclass.