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
Summer 2009

<table>
<thead>
<tr>
<th>Section/Problem</th>
<th>Points Earned</th>
<th>Points Possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Vocabulary Matching</td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>2. Fill in the Blank</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>3. Multiple Choice</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>4. Code Understanding</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>5. Turtle Graphics</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>6. Write Code: SumTo</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>7. Better Dorm</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>8. Convert Picture</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td><strong>Total Points:</strong></td>
<td></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>
1. Vocabulary Matching (32 points)

Write the number before the definition on the right on the line before the matching vocabulary word.

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Array</strong></td>
<td>A homogeneous linear collection of objects which are stored together in memory.</td>
</tr>
<tr>
<td><strong>Block</strong></td>
<td>A section of code, typically enclosed with curly brackets {} that makes up the body of a loop, function, or conditional.</td>
</tr>
<tr>
<td><strong>Boolean</strong></td>
<td>A data type that has only two possible values.</td>
</tr>
<tr>
<td>__Boolean Expression</td>
<td>A logical statement that evaluates to either True or False.</td>
</tr>
<tr>
<td>__Cast / Casting</td>
<td>The process of forcing Java to convert data from one data type to another.</td>
</tr>
<tr>
<td><strong>Class</strong></td>
<td>The fundamental building block of Java programs, they act as the blueprints from which objects are constructed, including definitions of fields and methods.</td>
</tr>
<tr>
<td><strong>Constructor</strong></td>
<td>A method that is called when a new instance is created.</td>
</tr>
<tr>
<td>__Data Encapsulation</td>
<td>The process of extending a superclass, gaining it's behaviors and state.</td>
</tr>
<tr>
<td><strong>Double</strong></td>
<td>A data type that holds numbers with fractional components.</td>
</tr>
<tr>
<td><strong>Field</strong></td>
<td>A data type that holds sequences of characters.</td>
</tr>
<tr>
<td><strong>Final</strong></td>
<td>A data type that has only two possible values.</td>
</tr>
<tr>
<td><strong>Graph</strong></td>
<td>A logical statement that evaluates to either True or False.</td>
</tr>
<tr>
<td><strong>Inheritance</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Int</strong></td>
<td>A keyword that means that only methods within the object can access a particular field or method.</td>
</tr>
<tr>
<td><strong>Iterate</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Linked List</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Method</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Method Signature</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Object</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Private</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Public</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Simula</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Simulation</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Smalltalk</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Static</strong></td>
<td>A keyword that indicates “nothing”, as in returns nothing.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td>A keyword that means that a field will not change.</td>
</tr>
<tr>
<td><strong>Subclass</strong></td>
<td>A function that is associated with an object and implements behavior.</td>
</tr>
<tr>
<td><strong>Superclass</strong></td>
<td>A homogeneous linear collection of objects which are stored together in memory.</td>
</tr>
<tr>
<td><strong>Traverse</strong></td>
<td>The unique collection of method name and parameter number and type that define a method. Multiple methods may share the same name as long as the type or number of their parameters differ.</td>
</tr>
<tr>
<td><strong>TypeDeclaration</strong></td>
<td>To (potentially) repeat the execution a block of code multiple times, as with a for or while loop.</td>
</tr>
<tr>
<td><strong>Void</strong></td>
<td>To move through a linear data structure (or a sequence) doing something to or with each individual element.</td>
</tr>
</tbody>
</table>
2. Fill in the Blank (12 points)

In Java, a __ = or single-equal sign ___ is used to indicate assignment, while a __ = = or double equal sign__ is used for equality checking.

Most statements in Java must be ended with a ___ ; or semi-colon ______________. The only exception is if you have only a single statement alone in a ___ block ______________.

In Java, logical and is written using the ___ && ______ symbol, and logical or is written using the ___ || ______ symbol.

Assume that the Student class is a subclass of the Person class, and the Person class is a subclass of the Human class. A variable that is defined to be of type Person can refer to (hold) an object of type __Person__________ or type _Student__________ but a variable defined to be of type Student can only refer to an object of type __Student__________.

A picture that is 200 pixels wide and 100 pixels high has a total of __200x100 or 20,000__ pixels. Each pixel needs __3_____ bytes, to represent the color of the pixel. How many total bits are used to represent the 200 by 100 picture? Answer: __200x100x3x8 or 480,000__

3. Multiple Choice (4 points)

Circle the correct answer.

Which of the following is an incorrect conditional statement?

A.  
   if ( thisColor == myColor )  
   setColor( thisPixel , newColor );

B.  
   if ( thisColor == myColor )  
   { setColor( thisPixel , newColor );  

C.  
   if ( thisColor == myColor )  
   {x = 12;  
   setColor ( thisPixel , newColor );  
   }  

CORRECT: D. All are correct.

The new operator:

A. invokes the constructor of an object.  
B. instantiates a specific instance of a class.  
C. allocates memory space for the object.  
D. A and C only.  
CORRECT: E. A, B, and C.
4. Code Understanding (9 points)

What does the following code output?

```java
int[] a = new int[10];
for (int i = 0; i < 10; i++) {
    a[i] = 9 - i;
}
for (int i = 0; i < 10; i++) {
    int index = a[i];
    a[i] = a[index];
}
for(int i = 0; i < 10; i++) {
    System.out.println(a[i]);
}
```

Answer: 0123443210

Grading:
- Exactly right = 9 points
- Numbers right, but printed horizontally = 8 points
- 1-2 numbers wrong = 6 points
- 3 or more wrong, but in ascending or descending order = 4 points
- Worse or blank = 3-0 points.

5. Turtle Graphics (6 points)

The following code creates a turtle and moves it around. Draw the turtle's path in the box on the right:

```java
Turtle t = new Turtle(new World());
for (int i = 0; i < 20; i++) {
    if (i % 2 == 0) {
        t.forward(10 * i);
    } else {
        t.turn(90);
    } // end if/else
} // end for
```

Grading:
- Turns in correct direction +2
- Forwards increasing by 20 each: +1
- First forward goes zero distance +1
- Correct number of lines/turns: +2
6. Write Code (12 points)

Write (twice) a method `sumTo` which returns the sum of the first \( n \) reciprocals. In other words, \( \text{sumTo}( n ) \) returns:

\[
1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + ... + 1/n
\]

You may assume that \( n \) is never a negative number. \( \text{sumTo}(0) \) returns 0.0, and \( \text{sumTo}(1) \) returns 1.0. The first time you write the method, use a for or while loop. The second time you write the method, you must use recursion instead of a loop. The return value and header for the method(s) are the same:

```java
// sumTo with a for or while loop
public static double sumTo(int n) {
    double answer = 0.0;
    for (int i = 1; i <= n; i++) {
        answer = answer + 1.0 / i; // or + (double) 1 / i;
    }
    return answer;
}
```

Grading:
6 pts total this part:
+1 points for returning answer as a double
+1 points for iterating through all the values from 1--n
+1 point for using casts or 1.0 to ensure correct division.
+1 point for summing all values together.
+2 got correct answer
-1 for minor syntax errors.

```java
// sumTo using recursion
public static double sumTo(int n) {
    if ( n == 0 ) {
        return 0.0;
    } else {
        double answer = 1.0 / n + sumTo(n-1);
        return answer;
    }
}
```

Grading: +2 points for returning correct answer. +1 point for correct 1.0 or cast to ensure double division. +2 points for correct recursive call. +1 correct terminating condition. -1 for minor syntax errors.
7. Create an Object: Better Dorm (15 points)

Examine the source code for the “Dorm” object provided at the end of this exam. Write code for a subclass of the Dorm object called BetterDorm. Your subclass must have:

- A constructor that accepts a string and two int's representing the name of the dorm and the number of men and women in the dorm.
- A method called percentMale that takes no parameters and returns the percentage of the dorm's population that is male (as a double). Your function must work for all possible numbers of men & women in the dorm.

```java
public class BetterDorm extends Dorm {
    public BetterDorm(String N, int M, int W) {
        super(N, M, W);
    }
    public double percentMale() {
        int M = getNumMen();
        int W = getNumWomen();
        if (M+W == 0) {
            return 0.0;
        }
        double answer = (double) M / (M+W);
        return answer; // or return answer * 100;
    }
} // end public class BetterDorm
```

Grading:
+2 for extending the superclass
+2 for the BetterDorm header correct
+2 – call to super in constructor correct.
+2 for the percentMale header correct
+2 for using accessor methods to get number of men/women.
+1 check for division by zero
+2 correctly calculating men / total (as double!)
+2 – returning correct answer
8. Convert a picture (10 points)

Write a new method for the Picture class that will return a black and white copy of itself. The method should make a new copy of itself, convert the pixels to B&W (using the following formula), and return it. Your method must be named `bwCopy`, take no parameters, and return a Picture object.

Use the following formula to convert from R,G,B values to a single Luminance (Y) value to put into all 3 (r,g,b) color slots:

\[ Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \]

```java
public Picture bwCopy() {
    int w = getWidth();
    int h = getHeight();
    Picture c = new Picture(w, h);

    Pixel[] orig = getPixels();
    Pixel[] copy = c.getPixels();

    for (int i = 0; i < orig.length; i++) {
        int R = orig[i].getRed();
        int G = orig[i].getGreen();
        int B = orig[i].getBlue();

        int Y = (int) (0.299 * R + 0.587 * G + 0.114 * B);

        copy[i].setRed(Y);
        copy[i].setGreen(Y);
        copy[i].setBlue(Y);
    } // end for each pixel.

    return c;
} // end bwCopy()
```

Grading:
+1 points for creating a new picture to save the Luminance data to.
+2 points for iterating through all pixels in the original picture (ourselves)
+2 points for retrieving the R,G,B values from the pixels.
+2 points for the Luminance calculation
+2 points for putting the Y data into the pixels in the copy.
+1 point for returning the B&W copy.
Definition of the “DORM” class:

public class Dorm {

    private String myName;
    private int myNumMen;
    private int myNumWomen;

    // Constructor
    public Dorm( String name, int men, int women) {
        myName = name;
        myNumMen = men;
        myNumWomen = women;
    }

    public void setNumMen( int men)
    {  myNumMen = men;  }

    public void setNumWomen( int women)
    {  myNumWomen = women;  }

    public void setName( String name)
    {  myName = name;}

    public int getNumMen()
    {   return myNumMen;  }

    public int getNumWomen()
    {   return myNumWomen;  }

    public String getName()
    {   return myName;  }

} // end dorm