CS 1322 Test 2 - A June 14, 2004

Name: _____

GT Number: _____

TA: _____

Commenting your code on this test is optional, however it may help the grader understand your code better. Other style issues such as using good variables names and using abstraction are still required. Write your gt number in the spaces provided and color in the corresponding numbers and letters.

gt	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	d	e	f	g	h	0	1	2	3	4	5	6	7	8	9	-	-	-	-	-	-	-	-	-	-	-
	0	1	2	3	4	5	6	7	8	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0	1	2	3	4	5	6	7	8	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	0	1	2	3	4	5	6	7	8	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Problem	Points Possible	Points Received	Graded By
1 Matching	20		
2 Accessors and Modifiers	13		
3 Interfaces	10		
4 Parameter Passing	12		
5 Applets	10		
6 Coding	10		
7 Tracing	10		
8 Linked List	15		
TOTAL	100		

Please remember: Any academic misconduct (including, but not limited to, the list below) could result in a 0 (zero) on the exam and/or an F grade in the course:

- Communication with anyone other than a proctor for ANY reason in ANY language in any manner.
- Sharing of ANYTHING (e.g. pencils, paper, erasers).
- Writing on paper that is not given to you by a proctor.
- Failure to follow directions given by the proctor.
- Use of cell phones, beepers, handheld computers, calculators, during the exam.
- Using books or other reference material.
- Disruption of the exam setting.

By taking this exam, you signify that it is your work and that you have neither given nor received inappropriate help during the taking of this exam in compliance with the Academic Honor Code of Georgia Tech.

Signature: _____

(you must sign this for your exam to be graded!)

1. Matching [20 pts]

Choose the **best** definition for each term.

1. _____ Applet
2. _____ Application
3. _____ alias
4. _____ private
5. _____ null
6. _____ this
7. _____ Constructor Chaining
8. _____ finalize
9. _____ garbage collection
10. _____ wrapper class

A. The use of this() to call another constructor in the same class

B. A Java reserve word that lets us convert a normal variable into a constant (final) variable

C. A feature of the Java Virtual Machine that allows it to automatically remove objects from memory when there is no valid reference to it remaining.

D. The condition when a reference variable contains the same reference as another, that is two variables refer to the same object.

E. A Java program that executes in stand-alone mode and has a main method

F. A Java reserved word used to create a constant value that cannot be modified

G. A Java reserved word which indicates a method in a class that can be called without requiring an object instance be created first

H. A class used to convert a primitive type into an object

I. A Java reserved word which indicates that a variable does not contain a valid object reference.

J. A feature of the Java Virtual Machine that allows it to remove objects from memory if the programmer has them implement the finalize interface.

K. A java reserved word that lets us instantiate new objects

L. A method which we can implement for a class, and is called when the object instance gets garbage collected.

M. A Java program designed to be run embedded in another program

N. A Java reserved word used to allow instance data to be used anywhere in an object, but prevent it from being used externally

2. **Accessors and Modifiers [13 pts]**

You are implementing a class Patron for a library. One piece of instance data is the patrons birth date which is of type Date. Date contains a static method public static int getAge(Date d) which given a Date will return an integer representing the number of years from that date to present time. Write the class definition for the Patron class and declare the instance data for birthday (3 pts). Then provide a constructor which takes in a Date and initializes the instance data (3 pts). Provide an method called getAge which returns the patron's age in years (as derived from their birthdate) (4 pts). Provide a modifier for the instance data birthday (3 pts).

3. Interfaces [10 pts]

4 (a) Define an interface named Cryptable which contains an integer constant MAX_KEY_SIZE which is 128. It should also require a method called encrypt which does not return anything and takes as a parameter an object of type Key.

6 (b) Define a class named Message which implements the Cryptable interface above. Message has an instance data element of type String named data. Write the class below. You may stub out any methods you need (i.e. the method body can be just { }).

4. Parameter Passing [12 pts]

```
public class FooBar {
  private int foo = 0;
  private String bar = "FOOBAR";
  public void check(int foo, String bar) {
    foo+=foo*10;
    bar=new String("OOPS");
  }//end method check

  public static void main(String[] args){
    FooBar fb = new FooBar();
    int foo=0;
    fb.check(7,fb.bar);
    fb.check(-6,fb.bar);
    System.out.println("The total is: "+fb.foo+" in "+ fb.bar);
  }//end main method

}//end class FooBar
```

The programmer expected the output to say: The total is: 10 in "OOPS".

[4] (a) Which of the following actually occurs with the code above:

1. The code compiles without error and produces the expected output.
2. The code compiles without error, but produces some other output.
3. The code causes a compiler error.

[4] (b) Explain your answer from above. If the program works correctly, explain why it does so. If the program produces incorrect output, include the actual output in your explanation. If the program will not compile, include why in your explanation.

[4] (c) If the program does not work properly, fix the code so that it works properly by clearly marking your changes above. Your fix may not be superficial (i.e. cross out all code and print the expected output). Your fix should address any issues explained above. If no changes are needed, then write "no changes" by code.

5. Applets [10 pts]

5 (a) If you want to draw a red circle inside green square in an applet where the paint method is passed a Graphics object called page, which of the following sets of commands might you use:

a)
```
page.setColor(Color.green);
page.fillRect(50,50,100,100);
page.setColor(Color.red);
page.fillOval(60,60,80,80);
```

b)
```
page.setColor(Color.red);
page.fillOval(60,60,80,80);
page.setColor(Color.green);
page.fillRect(50,50,100,100);
```

c)
```
page.setColor(Color.green);
page.fillRect(60,60,80,80);
page.setColor(Color.red);
page.fillOval(50,50,100,100);
```

d)
```
page.setColor(Color.red);
page.fillOval(50,50,100,100);
page.setColor(Color.green);
page.fillRect(60,60,80,80);
```

e) `any of the above will accomplish this.`

5 (b) The correct sequence of applet calls for an applet that is initially loaded, minimized and restored is:

a) `init(), start(), paint(), stop(), init(), start(), paint()`

b) `start(), init(), paint(), stop(), start(), init(), paint()`

c) `init(), start(), stop(), start()`

d) `init(), start(), paint(), stop(), start(), paint()`

6. Coding [10 pts]

You are to write a method which returns nothing called drawTriangle, which takes a single integer parameter named size. Based on the size, the method draws to the console an inverted triangle where spaces are shown by . and the row number is printed out the correct number of times. Your method should print the dots as shown in the output below. For example, assume we call drawTriangle with the parameter size = 4. The following should be the output:

```
4.4.4.4       HINT:  Look at the patterns.  (i.e. row n has n leading dots).
.3.3.3.
..2.2..
...1...
```

7. Tracing[10 pts]

What is printed by the following code.

```
public class Tester {
  private String name;
  private static int next=1;
  private int id;

  public Tester(String name) {
     this.name=name;
     this.id=next++;
     System.out.println(this);
  }//end constructor

  public Tester(){
     this("Bob");
     System.out.println("Made It");
  }

  public void check(int num){
     if (num==id)
        System.out.println("Found It");
     next++;
  }
  public String toString(){
     return("Name="+name+" ID="+id+" Next="+next);
  }
  public static void main(String[] args){
     Tester t1 = new Tester();
     t1.check(3);
     t1.check(1);
     Tester t2 = new Tester();
     t2.check(3);
     t1.check(1);
  }

}
```

Output:

8. **LinkedLists [15 pts]**

Code the method addAt which takes an integer index and an Object reference, and adds the object to the list at the given reference (zero indexed). The method should return a boolean which is true if the object is added and false otherwise. So addAt(0,name) would put the object referenced by name as the first element in the list and return true. addAt(4,name) would make the object be at the 5th position (0,1,2,3,4) and return true if there was at least 4 elements already in the array (something in positions 0,1,2,3) otherwise it would return false (and not insert anything). For instance if I try addAt(5,name) with an empty list, I return false because there is no such position in the list. If you successfully insert an element in the list, you should also update the size by 1. Assume you have the following LinkedList class partially implemented:

```
public class LinkedList {
  private Node head;
  private int size;

  private class Node {
     public Object data;
     public Node next;
     public Node(Object d) { data=d;next=null;}
  }
  /**the next two methods are implemented and you may use them*/
  public int size(){/*returns size of list*/}
  public boolean isEmpty(){/*returns true if the list is empty*/}
  //your addAt method would go here
}
```