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	10		
2	20		
3	10		
4	16		
5	8		
6	11 (+ 2 xtra)		
7	25		
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.
- Failure to stop writing when the allowed time is up (as reported 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. Vocabulary Terms – Short Answer

Define the terms listed below.

2 (a) object

2 (b) class

2 (c) constructor

2 (d) method

[2] (e) variable

2. Language Concepts – Short Answer

Provide a 2-3 sentence answer or explanation to the following:

5 (a) How does Java provide cross platform portability of code? Provide two or three specific facts or details.

5 (b) Describe the Java compilation and execution process, listing specific steps and commands.

[5] (c) List and describe two features that any object oriented programming language must have. What are these features called? Describe them.

[5] (d) Compare and contrast abstraction to encapsulation. How are they similar? How are they different?

3. Bug Parade

In the following fragments of code, identify the errors, based on the context given for each problem.

5 (a) The following code fails to compile. Identify the error, and rewrite the code fragment so that it works. (You may assume that the fragment appears in a proper class, and that all variables are properly declared; just focus on the fragment of code.)

```
public static void main (String[] arg);
{
      HelloWorld hw = new HelloWorld();
      hw.sayGreeting();
}
```

5 (b) The following code fragment was designed to print out 5 stars in a row, but when run, nothing happens. Identify the error or errors, and fix them. (As before, assume the code fragment appears in a proper class and method.)

```
int i = 0;
while (i <= 5);
{
   System.out.print('*');
   i++;
}
```

4. Assorted Language Features – Short Answer

[4] (a) List four signed data types (i.e., data types that are capable of holding positive and negative numeric values) in Java:

6 (b) Demonstrate (briefly) three techniques for commenting in Java.

6 (c) What is the minimum number of times the following loop constructs are *guaranteed* to execute:

i. A for loop

ii. A do-while loop

iii. A while loop

[8] 5. Code Fragments

[8] Rewrite the following lines of code so they do not use short-hand expressions. (Your answer may include more than one line.)

(a)
```
c++;
```

(b)
```
x/=5;
```

(c)
```
x-= (--x);
```

(d)
```
z = -(z--);
```

[11] 6. Casting and Data Types

[11] For each fragment below, identify whether the code would result in an error. If the code fragment is fine, write "OK" next to the problem. If the code fragment is not valid, write "ERROR", and rewrite the line so that it is correct, and accomplishes some useful purpose. (I.e., don't just comment out the code to fix it!) There may be many possible rewrites; however, you should not change the declared type of any variable. If you leave an answer blank, it will be marked wrong.

Hint: Each one is only worth 1 point! Don't waste time. Two of the problems (marked below) are extra credit.

(a)
```
int quickFuse = 42;
short temper = quickFuse;
```

(b)
```
float pi = 3.14159256;
```

(c)
```
short fall = 0000f;
```

(d)
```
long ssn = 123456789L;
int  y = (int) ssn;
ssn  = y;
```

(e)
```
short leave, shore = 10;
float boat = leave - shore;
```

(f)
```
short scheme = 5;
int x = ((int) ((long) ((short) ((char) ((int) scheme)))));
```

(g)
```
double trouble = 0L;
```

(h)
```
int steak = 2, bill = 3;
char broil = steak;
short change = bill;
```

(i)
```
boolean b = (1.0f > 3L);
```

(j)
```
int y = 4;
boolean b = ((long) y++ > 4f && --y < 10);
```

(k)
```
double vision = 2D;
```

[1] (l) **Extra Credit**

```
long contradiction = 0xC0B0L;
short andLong = contradiction;
```

[1] (m) **Extra Credit**

```
/* a floating pixie is one kind of elf */
float ingPixie = 1e1f;
```

25 7. Short Coding

15 (a) Complete the following method. The method below takes in a long, which has encoding for year, month, day, as follows: "YYYYMMDD". Thus, the number 20020529 is May 29, 2002. Variables called year, month and day are declared for you, and are printed out and the end of the method. You may not comment out or change any of the existing code.

```
public void printDates(long data){
  short year;
  byte  month, day;
  /* Your code goes here */









  System.out.println
    ("Year = " + year +
     ", month = " + month +
     ", day = " + day);
}
```

(You may use this page for your answer to the previous coding question).

10 (b) Rewrite the following code from a "for loop" to a "do...while" loop. The output should remain the same. That is, make sure your code behaves the same as the code below. You may omit the comments in your rewrite. As before, assume the fragment appears in a proper class and method.

```
/* print out odd numbers */
for (int i=0; i <= 8; i++){
  int remainder = i % 2;
  if (remainder != 0){
    System.out.println (i);
  }
}
```