CS 1322 Test 2 - A October 6, 2004

Name: _____

GT Number (email one): _____

TA and Section: _____

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 True/False | 10 | | |
| 2 Accessors/Modifiers | 16 | | |
| 3 Parameter Passing | 16 | | |
| 4 Array Coding | 30 | | |
| 5 Interfaces | 16 | | |
| 6 Tracing | 12 | | |
| 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!)

# EXAM 2 API Excerpt

Some, all or none of this API may be used on coding questions:

```
class Character
   constructor Character(char ch)  makes a new Character
   char charValue() returns the value as a char
   static int digit(char ch, int radix) returns the numeric
       value of the character ch in the specified radix
   static boolean isDigit(char ch)  returns true if ch is a digit
   static String toString(char ch) returns ch as a String object

class Integer
   constructor Integer(int i) makes a new Integer
   int intValue() return the int value of this Integer
   static int parseInt(String s) parses s and returns integer
   static String toString(int i) returns integer i as a String object

class String
   char charAt(int index)  returns the character at index, starting at 0
   int length() returns length of string
   int indexOf(String str) returns index where substring str found
   String[] split(String regex) break string on regular expression regex

class StringTokenizer
   constructor StringTokenizer(String s) make tokenizer on s
   constructor StringTokenizer(String s, String d)  make tokenizer
       on s using delimeters in d.
   int countTokens()  number of tokens found
   boolean hasMoreTokens() true if more tokens available
   String nextToken() return next token in tokenizer
```

1. **True/False [ 10 pts ]**

Answer the following questions True or False. Be sure to write out True and False, NOT just a T or F.

2 (a) The following statement will compile without error:
int [ ][ ][ ] arr = new int[5][5][ ];

2 (b) The following code snippet will compile without error:

```
public class Test{
   public int method1(String str){
     System.out.println(str);
     return str.length();
   }//method1
   public static void main(String[] args){
     String str;
     str = Keyboard.readString();
     int len = method1(str);
     System.out.println("The length of the string is:" + len);
   }
}
```

2 (c) JFrame and JPanel are the two top level containers for stand-alone and web-browser-based GUI programs.

2 (d) If your class implements the interface 'Comparable' it has to contain the method:
public int compareTo(Object o)

2 (e) For an array declaration **double arr**[][], the type of arr[0] is the double primitive.

## 2. Accessors(getters)and Modifiers(setters) [ 16 pts ]

Write a class named Coordinates. Coordinates should provide the following functionality:

1) Declare two instance variables called x and y with proper visibility to protect them from access outside the class.

2) Write a constructor that takes in two integer values and uses them to initialize the instance variables.

3) Write a constructor with no parameters that initializes x to 40 and y to 67 by chaining to the two parameter constructor.

4) Write an accessor method for the variable x and a modifier method for the variable y.

## 3. Parameter Passing [16 pts]

Consider the following ParamFun class:

```
public class ParamFun {
  public ParamFun(){
    System.out.println("Starting the fun. Are you with me?");
  }

  public void addToString(String myString){
     myString += "Hello";
  }

  public void replaceWith(char[] arr, char c){
     for(int i = 0; i < arr.length; i++){
        arr[i] = c;
     }
  }

  public void increaseByHundred(float num){
    num = num + 100;
  }

  public static void main(String [] Tucker){
    ParamFun pF = new ParamFun();
    String myString = "Yo ";
    char[] arr = {'a','b'};
    float num = 4.0f;
    pF.addToString(myString);
    pF.replaceWith(arr,'z');
    pF.increaseByHundred(num);
    System.out.println(myString);
    System.out.println(arr[0]);
    System.out.println(arr[1]);
    System.out.println(num);
  }
}//end class
```

When the program above is run, the programmer expects the output to be:

```
Starting the fun. Are you with me?
Yo Hello
z
z
104.0
```

8 (a) Write "YES" if the output is the same as expected by the programmer. Otherwise give the correct output.

8 (b) If you gave a different output explain why with reference to parameter passing concept in Java.

# 4. Array Coding [ 30 pts ]

15 (a) Code the method **parseIDNumber** which takes a String as parameter and returns an array of String. parseIDNumber should return the individual groups of numbers as separated by "-". Assume each group of digits separated by "-" is the same length.
For example, if parseIDNumber("4449-3789-2467") is called, the array returned should have three elements:"4449" "3789" and "2467".

15 (b) Write another method called **breakup**, that uses the method you wrote above. **breakup** takes a String as parameter and returns a 2D array of int. You should first call your method above to break the string into a String array. Then convert the String array into a 2D int array, where each string in the array forms a row and each digit in that string is a column. Note: Assume each String in the array returned from parseIDNumber will have the same length.
For example, if breakup("4449-3789-2467") is called, the 2D array returned is:
4 4 4 9
3 7 8 9
2 4 6 7

## 5. Interfaces[ 16 pts ]

8 (a) Write an interface called **MathFun**. This interface will contain one constant PI, which is equal to 3.14 and a public method called square, which takes in an int and returns an int.

8 (b) Write a class that implements your interface above. Your class must not produce any errors.

## 6. Tracing[ 12 pts ]

What is printed by the following code.

```
public class ACC {

  private static int GT=0;
  private int UGA=0;
  private int Miami=0;

  public ACC() {
    displayScores();
  }
  public void modifyScores() {
    int Miami=10;
    GT+=34;
    Miami++;
    UGA--;
  }
  public void displayScores() {
    System.out.println(GT+" "+Miami+" "+UGA);
  }
  public static void main(String[] Spring04){
    ACC DivisionA = new ACC();
    DivisionA.modifyScores();
    ACC DivisionB = new ACC();
    DivisionB.modifyScores();
    DivisionA.modifyScores();
    DivisionA.displayScores();
    DivisionB.displayScores();
  }
}
```

Output: