

CS 1331 Exam 1

Fall 2016

Name (print clearly): _____

GT account (gpburdell1, msmith3, etc): _____ Section (e.g., B1): _____

Signature: _____

- Failure to properly fill in the information on this page will result in a deduction of up to 5 points from your exam score.
- Signing signifies you are aware of and in accordance with the **Academic Honor Code of Georgia Tech** and that you will not discuss this exam with other students.
- Calculators and cell phones are NOT allowed.
- This is an object-oriented programming test. Java is the required language. Java is case-sensitive. DO NOT WRITE IN ALL CAPS. A Java program in all caps will not compile. Good variable names and style are required. Comments are not required.

Question	Points per Page	Points Lost	Points Earned	Graded By
Page 1	24	-	=	
Page 2	10	-	=	
Page 3	6	-	=	
Page 4	4	-	=	
Page 5	10	-	=	
Page 6	10	-	=	
Page 7	20	-	=	
Page 8	0	-	=	
TOTAL	84	-	=	

- [16] 1. Evaluate each of the expressions below. Write down the value that they evaluate to, and the resulting data type of that value in the provided columns. If the expression is not valid Java syntax, or will throw an exception, simply write "Error".

Table 1: Expressions

Expression	Resulting Value of Expression	Resulting Data Type
<code>(int)6.9 / 3</code>		
<code>(6 % 4 == 14 % 4)</code>		
<code>"cat" + "dog"</code>		
<code>"foo".equals(true ? "bar" : "foo")</code>		
<code>"I have " + 5 + " dogs"</code>		
<code>"Pickle".substring(3,5)</code>		
<code>"Pickle".charAt(5)</code>		
<code>"cat".indexOf("a")</code>		

2. **Legal or Not Legal** If the following lines of code would not generate any errors when compiled, write LEGAL and the resulting values of all listed variables. If it will generate a compilation error, write NOT LEGAL and explain the problem. Assume all variables are local variables.

[2] (a) `int i = 3;`
`double d;`
`d = 7 * d * 3;`

[2] (b) `float f;`
`double d;`
`d = 16.0;`
`f = (float) d / 2;`

[2] (c) `final int num = 99;`
`double num2 = 99.3;`
`num2 = num++;`

[2] (d) `long g = 12;`
`boolean b;`
`b = 12 - g;`

3. Short Answer

- [2] (a) Write the header for the method you need to define in a class to make it executable from the command line.
- [2] (b) Assume you are at the command line in the directory of the file that contains the definition for a Java class named **Avengers**. Write the command to compile **Avengers**.
- [2] (c) If the command above executes successfully, what file will be produced?
- [2] (d) Write the command that will execute the **Avengers** class you compiled above.
- [2] (e) What will the following code print?

```
for (int i = 2; i > 0; i--)  
    System.out.print("Go ");  
    System.out.print("Jackets!");
```

4. **Multiple Choice** Circle the letter of the correct choice.

Given:

```
public class Car {  
  
    public String model = "";  
    public int year = 0;  
  
    public Car(String aModel, int aYear) {  
        model = aModel;  
        year = aYear;  
    }  
    public String toString() {  
        return "Model: " + model;  
    }  
    public boolean equals(Object other) {  
        if (this == other) return true;  
        if !(other instanceof Car) return false;  
        if (this.model == ((Car) other).model) return true;  
        return false;  
    }  
}
```

Assume the following statements have been executed:

```
Car[] cars = { new Car("Camaro", 1967), new Car("Beetle", 1969), new  
                Car("Jaguar", 2015) };  
Car c = cars[1];  
c.model = "VW Beetle";
```

- [2] (a) What is the value of `c`?
- A. `null`
 - B. automatically set to 0
 - C. `cars[]`
 - D. the address of a `Car` object
 - E. None of the above
- [2] (b) What is the value of `cars.length`?
- A. `null`
 - B. the address of a `Car` object
 - C. 2
 - D. 3
 - E. None of the above
- [2] (c) What is the value of the expression `cars[1].equals(cars[2])`?
- A. `true`
 - B. `false`
 - C. This expression would cause an error to occur.

- [2] (d) What is printed on the console by `System.out.println(cars[1].toString());`?
- A. Model: Beetle Year: 1969
 - B. Model: VW Beetle Year: 1969
 - C. Model: Camaro Year: 1967
 - D. Car: null
 - E. None of the above
- [2] (e) What is printed on the console by `System.out.println(cars);`?
- A. null
 - B. ["Model: ", "Model: ", "Model: "]
 - C. ["Model: Camaro", "Model: Beetle", "Model: Jaguar"]
 - D. Something like [LCar;@1db9742]

5. Tracing

```
public class Tracer {
    public static void main(String[] args) {
        final String str = "fun 1331";
        for (int i = 0; i < str.length(); i++) {
            if ((i < 5) || (i > 2)) {
                System.out.print(str.charAt(i));
            } else {
                System.out.print(str.substring(i - 1, i));
            }
            if (i == 3) {
                System.out.println("Break time");
            }
        }
    }
}
```

- [5] (a) What is printed to the console when the program above is run?

```
class Tester {

    public static void main(String[] args) {
        String[] sa = new String[4];
        double[] da = new double[3];
        da[0] = 0;
        da[1] = 5;
        for (int i = 0; i < da.length; i++) {
            sa[i+1] = i + " : ";
            System.out.println(sa[i] + da[i]);
        }
    }
}
```

- [5] (b) What is printed when the program above is run?

[10] 6. Class and ObjectTracing

```
public class Poodle {

    private static int poodleCount = 0;
    private String name;

    public Poodle(String name) {
        this.name = name;
        poodleCount++;
    }
    public void setName(String newName) {
        name = newName;
    }
    public String getName() {
        return name;
    }
    public String toString() {
        return getName();
    }
    public String poodleCounter() {
        return "I am dog number " + poodleCount + ".";
    }
    public static void poodleReport() {
        System.out.println("There are " + poodleCount + " poodles in this kennel.");
    }
}

public class PoodleTester {
    public static void main(String[] args) {
        Poodle curly = new Poodle("Curly");
        System.out.println(curly.poodleCounter());
        Poodle chloe = new Poodle("Chloe");
        System.out.println(chloe.poodleCounter());
        Poodle jenny = chloe;
        jenny.setName("Jenny");
        System.out.println(jenny.poodleCounter());
        System.out.println(curly.poodleCounter());
        Poodle.poodleReport();
    }
}
```

What is printed when the PoodleTester is run?

7. **Write the code.** Be sure it would compile.

- [5] (a) Convert the following nested for loops to equivalent code that uses while loops instead instead of for loops. You do not need to rewrite the array declaration.

```
String[] [] arr = {"Hello", "there", "world"}, {"cs 1331", "is the", "best"};
for (int i = 0; i < arr.length; i++) {
    for (int j = 0; j < arr[i].length; j++) {
        System.out.print(arr[i][j]);
    }
    System.out.println();
}
```

- [15] (b) The morning of your CS1331 test, you decide to make yourself a healthy breakfast. You throw some food in the toaster. This isn't just any old toaster, though. It loves music, and it'll only toast if you let it sing along. You must create an object in Java to represent your toaster:
- * It has a specific number of slots for toast and other food items.
 - Add an instance variable to store this value.
 - This value is set when the object is instantiated, and the value cannot be changed afterwards.
 - * It has a specific song it likes to sing, which it remembers by name.
 - Add an instance variable to store the song name, and write methods to get and set this value. Be sure to use proper conventions for your getters and setters!
 - * Write constructors that allow the above two values to be set when creating the object.
 - One constructor should allow both values to be set using input parameters.
 - One constructor should set the number of slots to the default value of 1 and the song name to "Let It Be".
 - * Write a toast method which takes in the name of an item to toast.
 - This method should print out a string containing the song the toaster is singing and the item being toasted.
 - For example, if the song to sing is "Saint Pablo" and you're toasting bread, the string you print out could be "Singing Saint Pablo while I toast your bread!".

Write your complete Toaster class on the next page following the class heading. You do not need to write a tester for your Toaster class.


```
public class Toaster {
```