

# CS 1331 Exam 1A

## Spring Semester 2019 - February 1, 2019

Name: \_\_\_\_\_ Section: \_\_\_\_\_  
(print clearly including your first and last name)

Signature: \_\_\_\_\_

GT account username (msmith3, etc): \_\_\_\_\_

GT account number (903000000, etc): \_\_\_\_\_

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 is not allowed.
- You must have your BuzzCard or other form of identification on the table in front of you during the exam. When you turn in your exam, you will have to show your ID to the TAs before we will accept your exam. It is your responsibility to have your ID prior to beginning the exam.
- You are not allowed to leave the exam room and return. If you leave the room for any reason, then you must turn in your exam as complete.
- Signing and/or taking this exam signifies you are aware of and in accordance with the Academic Honor Code of Georgia Tech and the Georgia Tech Code of Conduct.
- Notes, books, calculators, phones, laptops, smart watches, headphones, etc. are not allowed.
- Extra paper is not allowed. If you have exhausted all space on this test, talk with your instructor. There are extra blank pages in the exam for extra space.
- Pens/pencils and erasers are allowed. Do not share ANYTHING.
- All code must be in Java.
- Style standards such as (but not limited to) use of good variable names and proper indentation is always required. (Don't fret too much if your paper gets messy, use arrows or whatever it takes to make your answer clear when necessary.) **If it is unclear which letters are capital and lowercase, underline the capital letters, ex: SomeName**
- Comments are not required unless a question explicitly asks for them.

Problem	Type	Points Possible
1) Foundation	Fill in the blank	6
2) Foundation	True/False	10
3) Foundation	Multiple Choice	10
4) Primitive Types	Ranking	4
5) Methods	Coding	12
6) Arrays	Coding	12
7) Strings	Short Answer	8
8) Switches	Tracing	8
9) Math Expressions	Short Answer	10
10) Boolean Expressions	Short Answer	10
11) Iteration	Coding	10
Total Points		100
Bonus question 1		2
Bonus Question 2		1

# 1) [6 pts] Foundation - Fill in the blank

For each of the questions below, fill in the bubble corresponding to the correct word from the word bank.

Word Bank	
A	reserved word
B	machine code
C	source code
D	bytecode

A	B	C	D	
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	[2pts] The Java Compiler (javac) generates _____, which is run by the JVM and makes a Java program hardware agnostic.
<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	[2pts] A _____ has a specific meaning to the compiler and cannot be used for other purposes in the program.
<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	[2pts] _____ is human readable and is what the software programmer writes.

## 2. [10pts] Foundation - True/False

True	False	
<input type="radio"/>	<input checked="" type="radio"/>	[2pts] In Java, String objects are mutable.
<input checked="" type="radio"/>	<input type="radio"/>	[2pts] Java is case sensitive.
<input type="radio"/>	<input checked="" type="radio"/>	[2pts] In Java, constants are indicated with the <code>const</code> term.
<input checked="" type="radio"/>	<input type="radio"/>	[2pts] Different types of numeric values can be used together in a computation.
<input checked="" type="radio"/>	<input type="radio"/>	[2pts] In Java, you can compare characters with relational operators.

## 3. [10pts] Foundation - Multiple Choice

a. [2pts] When comparing two variables of type *int* *a* and *b*, which techniques are **valid**?

- ☒ `a == b`
- ☐ `a += b`
- ☐ `a.equals(b)`
- ☐ both `a == b` and `a.equals(b)` are valid

b. [2pts] Given the following enum definition, which of the statements is **NOT valid**?

```
public enum Colors {RED, BLUE, YELLOW}
```

<input type="radio"/>	<code>Colors myColor = Colors.RED;</code> <code>myColor.name();</code>
<input checked="" type="radio"/>	<code>Colors myColor = Colors.BLUE;</code> <code>myColor.length;</code>
<input type="radio"/>	<code>Colors myColor = Colors.YELLOW;</code> <code>myColor.ordinal();</code>
<input type="radio"/>	<code>System.out.println(Colors.BLUE);</code>

c. [2pts] Given two doubles **a** and **b**, how do you perform the operation **a<sup>b</sup>**

**a<sup>b</sup>**  
pow

- ☐ `a ** b`
- ☐ `a.pow(b)`
- ☐ `a ^ b`
- ☒ `Math.pow(a, b)`

d. [2pts] Given a variable of type **Scanner** called **scan**. Which of the following does **NOT** work?

- ☐ `scan.nextInt();`
- ☐ `scan.nextLine();`
- ☒ `scan.all();`
- ☐ `scan.next();`

e. [2pts] How do you properly escape a *newline* character?

- ☐ `\br`
- ☒ `\n`
- ☐ `{newline}`
- ☐ `/br`

#### 4. [4pts] Primitive Types

Rank the following primitive Integer types from the smallest storage size to the largest storage size (1 being smallest and 4 being the largest). Only select 1 circle for each row:

1 smallest	2	3	4 largest	
<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	int
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	long
<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	byte
<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	short

## 5. [12 pts] Methods

[8 pts] Part 1 - Write a method that overloads the sayHi method below to take in a single `int` variable parameter called `count`. The method you write should print out "Hi X!" the number of times passed in via the `count` parameter, where X is your name.

For example, for me, `sayHi(2)` would print:

"Hi Suzy!"

"Hi Suzy!"

**Do not duplicate the code provided! Reuse code whenever possible!**

[4 pts] Part 2 - Draw a box around the **method signature** for each method.

```
public class MethodFun {  
    public static void main(String[] args) {  
        sayHi("Hello", "Suzy", 2); // prints "Hi Suzy!" 2 times  
        sayHi(5); // prints "Hi <your name>!" 5 times  
    }  
  
    public static void sayHi(String prefix, String name, int count) {  
        for (int i = count; i>0; i--) {  
            System.out.println(prefix + " " + name + "!");  
        }  
    }  
}
```

```
public static void sayHi(int count) {  
    sayHi("Hi", "Suzy", count);  
}
```

## 6. [12 pts] Arrays

Declare and instantiate an array called `myArray`. Then use a *loop* to initialize the array with the first 10 even numbers starting at 12. In other words, have even numbers from 12 to 30 in the array. DO NOT use the shorthand notation for initializing an array.

`myArray`:

12	14	16	18	20	22	24	26	28	30
----	----	----	----	----	----	----	----	----	----

```
public class ArrayFun {
```

```
    public static void main(String[] args) {  
        // Your code below:
```

```
        int myArray = new int[10];  
        for (int i = 0; i < myArray.length; i++) {  
            myArray[i] = (12 + (2 * i));  
        }
```

OR

```
        int myArray = new int[10];  
        for (int num = 12; num <= 30; num += 2) {  
            myArray[num / 2 - 6] = num;  
        }
```

```
    }
```

## 7. [8 pts] Strings

Given the following block of code, what is printed out?

```
String a, b, c, d, e;  
a = "Computer Science";  
b = new String("1331");  
c = a.replace('C', 'd'); → "domputer Science"  
d = a.substring(0,3); → "Com"  
e = a.substring(a.length()-7, a.length()-4); → a.substring(9,12) → "Sci"
```

- A) System.out.println(a + " - " + b);
- B) System.out.println(c + " - " + b);
- C) System.out.println(d + e + " - " + b);

A) Computer Science - 1331

B) domputer Science - 1331

C) Com Sci - 1331



## 8. [8pts] Code Tracing

Given the following code, what is the output.

```
public static void main(String[] args) {
    System.out.print("salad? ");
    evaluateFood("salad");
    System.out.print("CHOCOLATE? ");
    evaluateFood("CHOCOLATE");
    System.out.print("Veggies? ");
    evaluateFood("Veggies");
    System.out.print("candy? ");
    evaluateFood("candy");
}

public static void evaluateFood(String food) {
    switch(food) {
        case "chocolate":
            System.out.println("My absolute favorite!");
        case "candy":
            System.out.println("Yum!");
            break;
        case "salad":
        case "Veggies":
            System.out.println("Only if I have to :-)");
            break;
        case "Pizza":
        default:
            System.out.println("Sure, I'll try it...");
    }
}
```

Write the output here:

Line 1:	Salad? Only if I have to :-)
Line 2:	CHOCOLATE? Sure, I'll try it...
Line 3:	Veggies? Only if I have to :-)
Line 4:	Candy? Yum!

## 9. [10 pts] Math Expressions

What are the values of the following expressions? Assume that the integer x has the value of 2. Carefully consider whether to include a decimal point in your answer.

a)  $2 * 3 / 5 - 4$

$$\Rightarrow 6/5 - 4 = 1.2 - 4 = -2.8$$

Answer: -2.8

b)  $(11 \% x) * 4.3$

$$\Rightarrow 1 * 4.3$$

Answer: 4.3

c)  $(\text{int}) (4.4 * x)$

$$(\text{int}) (4.4 * 2) = (\text{int}) (8.8) = 8$$

Answer: 8

d)  $10 \% 3 - 5 * 2$

$$1 - 10 = -9$$

Answer: -9

e)  $(\text{double}) x / 4$

$$2.0 / 4 = 2.0 / 4.0 = 0.5$$

Answer: 0.5

## 10. [10 pts] Boolean Expressions

Assume that the integer variable y has a value of 3. Are the following boolean expressions true or false? [2pt each]

True	False	
<input checked="" type="radio"/>	<input type="radio"/>	$(y \neq 3) \    \ (y < 5)$ false    true
<input checked="" type="radio"/>	<input type="radio"/>	$\text{true} \ \&\& \ (y > 0)$ true & true
<input checked="" type="radio"/>	<input type="radio"/>	$(y \leq 3) \    \ (y \leq 0)$ true
<input type="radio"/>	<input checked="" type="radio"/>	$!(y == 3)$ !true
<input checked="" type="radio"/>	<input type="radio"/>	$(y \neq 7) == (y == 3)$ true == true

## 11. [10 pts] Iteration

Convert the following code into a for-loop.

```
int count = 10;  
do {  
    System.out.println("count is: " + count);  
    count--;  
} while (count > 0);
```

*← at least 1x*

Write your answer here:

```
for (int count = 10; count > 0; count--) {  
    System.out.println("count is: " + count);  
}
```

### Bonus #1 [2 pts]

Think about what this ? is asking you to know.

What is the output of the following if  $x = 1$ ,  $y = -1$  and  $z = 0$ :

```
if (x > 0)
    if (y > 0)
        System.out.println("x > 0 and y > 0");
    else if (z > 0)
        System.out.println("x <= 0 and z > 0");
    else
        System.out.println("x <= 0 and z <= 0");
```

### Bonus #2 [1 pt]

Name a non-numeric (not a number) **primitive** Java type.