



# University of Maryland College Park

## Department of Computer Science

### CMSC131 Spring 2023

#### Exam #1

FIRSTNAME, LASTNAME (PRINT IN UPPERCASE):

KEY

STUDENT ID (e.g. 123456789):

#### Instructions

- Please print your answers and use a pencil.
- Do not remove the staple from the exam. Removing it will interfere with the Gradescope scanning process.
- To make sure Gradescope can recognize your exam, print your name, write your directory id at the bottom of pages with the text DirectoryId, **provide answers in the rectangular areas provided**, and do not remove any exam pages. Even if you use the provided extra pages for scratch work, they must be returned with the rest of the exam.
- This exam is a closed-book, closed-notes exam, with a duration of 50 minutes and 100 total points.
- Your code must be efficient.
- Multiple choice questions have only one answer unless indicated otherwise.
- You don't need to use meaningful variable names; however, we expect good indentation.

#### Grader Use Only

#1	Problem #1 (Short Answers -2pts each)	16
#2	Problem #2 (Short Answers -3pts each)	18
#3	Problem #3(Code 1)	28
#4	Problem #4 (Code 2)	38
<b>Total</b>	Total	100

## Problem #1 (Short Answers – 2 pts)

1. (2 pts) Which statement is false?
  - a. "A" is a `String` literal and not a primitive character.
  - b. A local variable of type `int` will be implicitly assigned 0.
  - c. An example of a Java primitive type is the type `boolean`.
  - d. `=` is the assignment operator in Java.
2. (2 pts) Which statement is true?
  - a. You make a block comment in Java using `//`
  - b. Forgetting to terminate a Java statement would be an example of a runtime error
  - c. A `String` is one of Java primitive types
  - d. You cannot have a variable in Java called `if`.
3. (2 pts) A unit of code that performs a task when invoked, is the definition of a:
  - a. Literal
  - b. Method
  - c. Class
  - d. Variable
4. (2 pts) The base 10 number 156 is what in octal (base 8) (no calculators)?

234

5. (2 pts) Assume the following code fragment in a `main` method, but it still will not compile. Why?

```
int x;  
int num = 8;  
int result = x + num;  
System.out.println(result);
```

x has no initial value and therefore cannot be used on the right side of the assignment statement

6. (2 pts) The expression `"Ben".equals("ben")` returns:
  - a. A negative integer
  - b. A positive integer
  - c. Zero
  - d. false
  - e. true
  - f. None of the above.
7. (2 pts) Assume the following code fragment in a `main` method, what is the output? \_\_\_\_\_5\_\_\_\_\_

```
int x = 5;  
if ("hi".compareTo("hi")!=0)  
    x = 10;  
if (false && ++x > 0)  
    x++;  
else if (x > 5)  
    x = 30;  
System.out.println(x);
```

8. (2 pts) Assume the following code fragment in a main method, how many times will hi print? Answer needs to be a number.

```

for(int i= 1; i<=57; i++) {
    System.out.println("hi");
    for(int j=22; j>22; j++){
        System.out.println("hi");
    }
    if(i==3)
        i=17;
}

```

43

**Problem #2 (Short Answers – 3 pts)**

9. (3 pts) Name all three looping construct we have covered in class.

**1. for**

**2. while**

**3. do while**

10. (3 pts) Show how you would declare in the main method a constant variable of the type int called ID and assign to it 347. Just write the one line statement, you don't need to write out the header for main.

final int ID =347;

11. (3 pts) Complete the truth table for the and operator (i.e. &&). That is **Result = Condition 1&& Condition 2**

Condition 1	Condition 2	Result
False	False	False
False	True	False
True	False	False
True	True	True

12. (3 pts) Assume the variable, myStr, has a string assigned to it. Show how you would concatenate the string to its uppercase version and assign it back to myStr. For example, if the original String is hello, after your code runs the new String assigned to myStr should be helloHELLO. Just write the assignment statement, no need for a main method.

myStr +=myStr.toUpperCase();

13. (3 pts) Assume the following code fragment in a main method. Simply write down what the three print statements will print.

```

int x =10,y;
System.out.println(x--);
System.out.println(++x > 5);
System.out.println(y=x--);

```

10  
true  
10

Directory id:

14. (3 pts) The `ceil` method takes a `double` argument and returns the ceiling of the value. It is a `static` method of the `Math` class. Show how you would call it, passing in the expression `90.0/100`, and printing out the value using the `println` method. You must do it all in one Java statement .

```
System.out.println(Math.ceil(90.0/100));
```

### Problem #3 (Code 1)

Write a public static method called `makeStr` which returns a `String` and has one `String` parameter: `input`. If the first character of the argument is an A or B, return the same string as the argument with an `*` before and after it. If the first character of the argument is a C and the last character is a d, return the same string as the argument, followed by `*`, followed by the same string as the argument. If the first character of the argument is an E, return the same string as the argument in all uppercase. For any other argument, return a string that is the reverse of the argument string.

In your code, you may use the following `String` library methods: `charAt`, `length`, and `toUpperCase`. You need to write the code logic to reverse the string and cannot try to call a method that does all that work for you. If you have a `System.out.println` in your code, you are not writing the code correctly. No need for a `main`. The code logic should be case sensitive.

Sample calls	Output of the calls
<code>System.out.println(makeStr("Apple"));</code>	<code>*Apple*</code>
<code>System.out.println(makeStr("Bread"));</code>	<code>*Bread*</code>
<code>System.out.println(makeStr("bread"));</code>	<code>daerb</code>
<code>System.out.println(makeStr("Car"));</code>	<code>raC</code>
<code>System.out.println(makeStr("Card"));</code>	<code>Card*Card</code>
<code>System.out.println(makeStr("Eyes"));</code>	<code>EYES</code>
<code>System.out.println(makeStr("Java"));</code>	<code>avaJ</code>

```
public static String makeStr(String input) {  
  
    if(input.charAt(0)=='A' || input.charAt(0)=='B')  
        return "*" + input + "*";  
    else if (input.charAt(0)=='C' && input.charAt(input.length()-1)=='d')  
        return input + "*" + input;  
    else if (input.charAt(0)=='E')  
        return input.toUpperCase();  
    else  
    {  
        String temp = "";  
        for (int i =input.length()-1; i >=0; i--)  
        {  
            temp+=input.charAt(i);  
        }  
        return temp;  
    }  
  
}
```

//More space for code 1 problem if you need it

## Problem #4 (Code 2)

Write a public static method called `makeFibTriangle` which returns a `String` and has two parameters: an `int` parameter called `rows` and a `char` parameter called `myChar`. If `rows` is larger than 12 or smaller than or equal to 0, simply return the string `:(`  
Otherwise your code will return a `String` of a triangle with the number of rows being the parameter `rows` and that alternates between terms of the Fibonacci sequence and `myChar` (always starting with the term 0 of the Fibonacci sequence). The Fibonacci sequence starts with 0 and 1 and the next term is derived from the sum of the two previous terms. For example, the first six terms are 0,1,1,2,3,5. The third term, 1, is the sum of 0 and 1. The fourth term, 2, is the sum of 1 and 1. The fifth term, 3, is the sum of 1 and 2, and so on. If you have a `System.out.println` in your code, you are not writing the code correctly. No need for a `main`.

Sample calls	Output of the calls
<code>System.out.println( makeFibTriangle(-10, '%'));</code>	<code>:(</code>
<code>System.out.println( makeFibTriangle(13, '\$'));</code>	<code>:(</code>
<code>System.out.println( makeFibTriangle(10, '%'));</code>	<code>0</code> <code>% 1</code> <code>% 1 %</code> <code>2 % 3 %</code> <code>5 % 8 % 13</code> <code>% 21 % 34 % 55</code> <code>% 89 % 144 % 233 %</code> <code>377 % 610 % 987 % 1597 %</code> <code>2584 % 4181 % 6765 % 10946 % 17711</code> <code>% 28657 % 46368 % 75025 % 121393 % 196418</code>

```
public static String makeFibTriangle(int rows, char myChar){

    if (rows >12 || rows<=0)
        return ":( ";
    else
    {

        String result ="";
        int prev =0, current =1, oldPrev;
        result+=prev+"\n";
        boolean flag =false;

        for (int row = 2; row <= rows; row++) {

            for (int col = 1; col <= row; col++) {

                if(flag)
                {
                    result+=current+ " ";
                    oldPrev = prev;
                    prev = current;
                    current+=oldPrev;
                }
                else
                {
                    result+=myChar+ " ";
                }
                flag =!flag;
            }

            result+="\n"; // Next line
        }

        return result;
    }
}
```

**//More space for code 2 problem if you need it**

**Directory id:**