



# University of Maryland College Park

## Department of Computer Science

### CMSC131 Spring 2022

### 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)	24
#3	Problem #3(Code 1)	30
#4	Problem #4 (Code 2)	30
<b>Total</b>	Total	100



## **Problem #1 (Short Answers – 2 pts)**

1. (2 pts) Which statement is false?
  - a. Java compiles to byte code.
  - b. Eclipse is an IDE.
  - c. There are 9 Java primitive types.
  - d. Java is an object oriented programming language.
2. (2 pts) Which statement is true?
  - a. The code you write in Java is called the object code, not source code.
  - b. A method is Java's term for what is sometimes called a procedure, subroutine, or a function.
  - c. Declaring a variable is not an example of a Java statement.
  - d. In Java you can have a main method outside of a class definition.
3. (2 pts) The name of a location in memory that holds data that can change, is the definition of a:
  - a. Literal
  - b. Method
  - c. Class
  - d. Variable
4. (2 pts) The base 10 number 171 is what in hexadecimal (base 16) (no calculators)?

**AB**

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

```
int x = 7;  
int num = 8;  
int result = x + Num;  
System.out.println(result);
```

**Case sensitive, no variable called Num**

6. (2 pts) The expression "Ben".compareTo("Bill") returns:

- a. A negative number
- b. A positive number
- c. Zero
- d. None of the above.

7. (2 pts) Assume the following code fragment in a main method, what is the output?

```
int x = 5;  
if (x != 7)  
    x = 10;  
if (x > 5)  
    x = 20;  
else  
    x = 30;  
    x = 31;  
System.out.println(x);
```

**31**

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++)  
{  
    for(int j=1; j<22; j++)  
    {  
        System.out.println("hi");  
    }  
}
```

**1176**

**Directory id:**

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

9. (3 pts) Assume the declared variable `num` has been assigned a positive odd integer. Use a **while loop** to show how you would assign to the variable `sum`, the sum of all odd integers from 1 up to and including `num`. For example, if `num` is 5, `sum` would be 9 (i.e.  $1 + 3 + 5$ ). Just write the code for the loop, no `main` needed.

```
int num =5; //your code should work even if num is another positive odd other than 5.
int i =1;
int sum =0;
while(i<=num)
{
    sum+=i;
    i+=2;
}
```

```
System.out.println(sum);
```

10. (3 pts) Use a **do while** loop to print CMSC131, 131 times. Just the code for the loop, no `main` needed. The abbreviated output is below. **[And so on]** will not print out, the actual text will. Notice, you **do need** to print the number and :

```
1: CMSC131
2: CMSC131
    [And so on]
130: CMSC131
131: CMSC131
```

```
int i = 1;
do {
    System.out.println(i+ ": CMSC131");
    i++;
}while (i<=131);
```

11. (3 pts) Name any three Java primitive types other than `int` or `double`.

**Any three of char, byte, boolean, short, long or float**

12. (3 pts) Declare a named constant of the type `double` called `PI` and assign to it 3.14.

**`final double PI = 3.14;`**

13. (3 pts) Complete the truth table for the or operator (i.e. `|`).

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

14. (3 pts) Assume the variable, `myStr`, has a string assigned to it. Show how you would concatenate the string to its length 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 `hello5`. Just write the assignment statement, no need for a `main` method.

**`myStr += myStr.length();`**

15. (3 pts) Assume the following code fragment in a `main` method. Simply write down what the three `println` statements will print.

```
int x = 7, y;  
System.out.println(x++); //7  
System.out.println(++x); //9  
System.out.println(y++x); //10
```

16. (3 pts) The `abs` method takes an integer argument and returns the absolute value of the argument. It is a static method of the `Math` class. Show how you would call it, passing in -17, and assigning the return value as the initialization value to a variable called `ans`. You must do it all in one Java statement.

**`int ans = Math.abs(-17);`**

**Directory id:**

### Problem #3 (Code 1)

Just add a static method, called `exam1Method`, to the class below which returns an integer and has three integer parameters: `num1`, `num2`, and `num3`. You can assume that integers will be passed in when the method is called. If at least one of the integers is less than 0, simply return the sum of the three parameters. Otherwise, check to see if all the parameters are even. If so, return the product of the three parameters. If the parameters passed in, have not satisfied the previous 2 requirements, simply return the smallest value out of the three. If you have a `System.out.println` in your code, you are not writing the code correctly. No need for a `main`.

<code>System.out.println(exam1Method(5,-20,8));</code>	<code>//at least one negative, so returns sum</code>	-7
<code>System.out.println(exam1Method(0,4,18));</code>	<code>///all non-negative evens, returns product</code>	0
<code>System.out.println(exam1Method(8,4,6));</code>	<code>///all non-negative evens, returns product</code>	192
<code>System.out.println(exam1Method(10,13,10));</code>	<code>///returns smallest out of the three</code>	10
<code>System.out.println(exam1Method(57,18,7));</code>	<code>///returns smallest out of the three</code>	7

```
public class P3 {
```

```
    public static int exam1Method(int num1, int num2, int num3)
    {
        if (num1 < 0 || num2 < 0 || num3 < 0)
            return num1+num2+num3;
        //you know they are all 0 or positive
        else //not needed due to return
        {
            if(num1%2==0 && num2%2==0 && num3%2==0) //all non-negative evens
                return num1*num2*num3;
            else //not needed due to return
            {
                if (num1 < num2 && num1 < num3)
                    return num1;
                else if (num2 < num3)
                    return num2;
                else
                    return num3;
            }
        }
    }
}
```

## Problem #4 (Code 2)

Complete the implementation of the program below that will prompt for the number of rows. Assume a positive `int` as the input. The output will be the row number, followed by a colon, followed by the same number of symbols as the row number, followed by an equal sign, followed again by the row number. As for the symbol, use `?` if the row number is divisible by 3, otherwise, use `*`.

A sample run is shown on the right (with user input in italic and output in bold just to make it easier to understand how it works).

**Enter number of rows:** *10*

**1:\*=1**  
**2:\*\*=2**  
**3:??=3**  
**4:\*\*\*\*=4**  
**5:\*\*\*\*\*=5**  
**6:?????=6**  
**7:\*\*\*\*\*=7**  
**8:\*\*\*\*\*=8**  
**9:???????=9**  
**10:\*\*\*\*\*=10**

```
import java.util.Scanner;
public class P4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter number of rows: ");
        int rows = scanner.nextInt();

        for (int i = 1; i <= rows; i++)
        {
            System.out.print(i + ":");
            for (int j = 1; j <= i; j++)
            {
                if(i%3!=0)
                    System.out.print("*");
                else
                    System.out.print("?");
            }
            System.out.print("=" + i);
            System.out.println();
        }
    }
}
```

**EXTRA PAGE IN CASE YOU NEED IT**

**LAST PAGE**