

Name: _____	CSCI 6101 OOP Using Java Georgia Southern University Instructor: Dr. Y. Daniel Liang
-------------	--

I pledge by honor that I will not discuss this exam with anyone during the exam.

Signed by _____ Date _____

This is an open book exam. You may reference the book during the exam.

Your programs will be submitted to LiveLab. Creating a LiveLab account is easy. Follow the steps below to create LiveLab accounts.

- Go to <http://livelab.georgiasouthern.edu/JavaLiveLab2020>
- Under the Student tab, click Create Account
- Use Signup Code **coding101** and Course ID **csci6101fall2021** to create your account. Use your last name and underscore (_) and first name for username. For example, my name is Yong Liang, the username is liang_yong. **Choose a password that is unique to this Website. Don't use this password on other Websites. (THIS IS IMPORTANT FOR SECURITY. PLEASE FOLLOW THIS INSTRUCTION.)**

After you submit your code, log out and log in back to verify your submission.

1. (10 pts) Write a method that sums all the numbers in the major diagonal in an $n \times n$ matrix of `double` values using the following header:

```
public static double sumMajorDiagonal(double[][] m)
```

Write a test program that reads a 4-by-4 matrix and displays the sum of all its elements on the major diagonal. Here is a sample run:

<output>

Enter a 4-by-4 matrix row by row:

1 2 3 4.0 *<enter icon>*

5 6.5 7 8 *<enter icon>*

9 10 11 12 *<enter icon>*

13 14 15 16 *<enter icon>*

Sum of the elements in the major diagonal is 34.5

<end output>

Your code should look like this:

```
public class PFinal1 {  
    public static void main(String[] args) {  
        // WRITE YOUR CODE  
  
    }  
  
    public static double sumMajorDiagonal(double[][][] m) {  
        // WRITE YOUR CODE  
    }  
}
```

2. (10) Write a class named Clock that meets the following requirements:

The class has three instance variables: One of type int called hours, another of type boolean called isTicking, and the last one of type Integer called diff.

The class has a constructor that takes three parameters: an int for hour, a boolean for isTicking, and another int for diff. The constructor constructs a Clock with the specified values.

A method named increment with no arguments and no return value. It increases the hours by 1 if the hour is less than 12.

A method named decrement with no arguments and no return value. It decreases the hours by 1 if hour is greater than 0.

Override the toString() method that returns a string. The string is

```
"hours: " + hours + " isTicking: " + isTicking + " " +
isTicking + " diff: " + diff
```

Write a test program to perform the following operations:

1. Prompt the user to enter an integer for hours, a Boolean value for isTicking, and an integer for diff.
2. Create a Clock object using the input values.
3. Display the object by invoking its toString() method.
4. Invoke the increment method twice and then the decrement method once.
5. Display the object by invoking its toString() method.

Your code should look like this:

```
public class PFinal2 {

    public static void main(String[] args) {

        // WRITE YOUR CODE

    }

}
```

```
Class Clock {

    // WRITE YOUR CODE

}
```

Here is a sample run:

<output>

Enter an integer for hours: 10

Enter a Boolean value for isTicking: true

Enter an integer for diff: 5

hours: 10 isTicking: true diff: 5

hours: 11 isTicking: true diff: 5

<end output>