Seneca College

Nov 16, 2019

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444 **Demo & Code Submission Date: Nov 22, 2019**

Workshop 8

Notes:

i. Each task should be presented during the lab, demo worth 70% of the workshop marks and code uploading worth the other 30%.
ii. Make sure you have all security and check measures in place (with proper use of Exceptional Handling where ever needed), like wrong data types etc.
iii. Make your project in proper hierarchy; introduce proper class coherence in your project. Proper packages and **your project should be handled by only one main method which should be in a TesterClass**.
iv. Given output structure is just for student to have a glimpse what the output can look, students are free to make the output better in any way.

Other inputs can be given during demo, so make sure you test your program properly.

Task 1 (Lambda Practice):

This exercise asks you to write a few lambda expressions and a function that returns a lambda expression as its value. Suppose that a functional interface *ArrayProcessor* is defined as

```
@Functional Interface
public interface ArrayProcessor {
   double apply( double[] array );
}
```

Write a class that defines four *public static final* variables of type ArrayProcessor that process an array in the following ways:

1. find the maximum value in the array
2. find the minimum value in an array
3. find the sum of the values in the array
4. find the average of the values in the array.

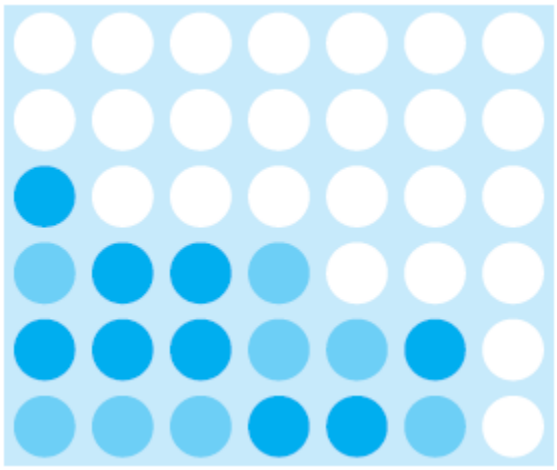
In each case, the value of the variable should be given by a lambda expression. The class should also define a function

public static ArrayProcessor counter(double value) { ...

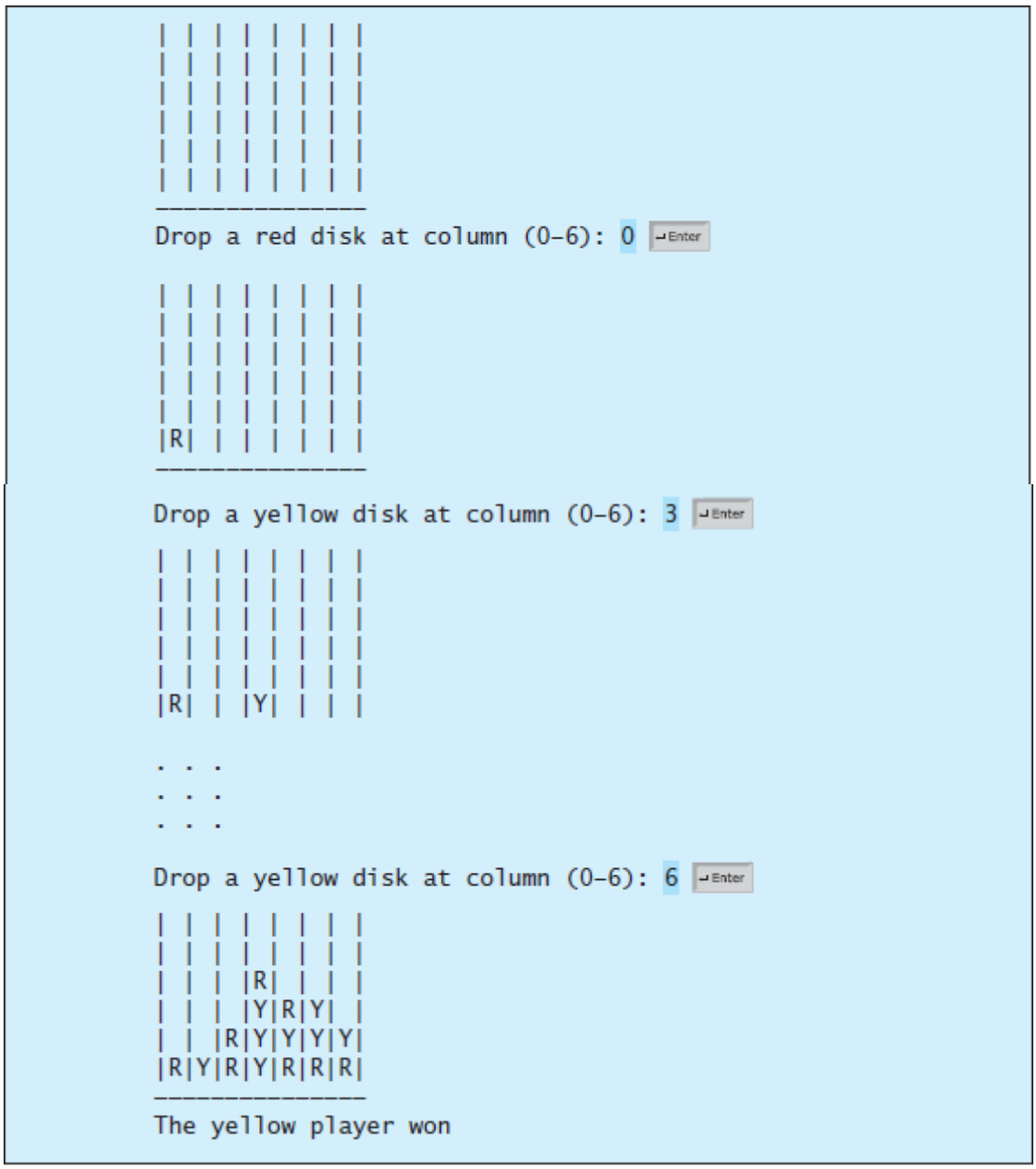
This function should return an ArrayProcessor that counts the number of times that *value* occurs in an array. The return value should be given as a lambda expression.

The class should have a *main*() routine that tests your work. Ask the user to give array elements.

Task 2: (Game – Connect four) (Students can implement the game with JavaFX or without) Connect four is a two-player board game in which the players alternately drop colored disks into a seven-column, six-row vertically suspended grid, as shown below.



The objective of the game is to connect four same-colored disks in a row, a column, or a diagonal before your opponent can do likewise. The program prompts two players to drop a red or yellow disk alternately. In the preceding figure, the red disk is shown in a dark color and the yellow in a light color. Whenever a disk is dropped, the program redisplays the board on the console and determines the status of the game (win, draw, or continue). Here is a sample run:



Drop a red disk at column (0–6): 0 Enter
R						

Drop a yellow disk at column (0–6): 3 Enter
R			Y			
. . .
. . .
. . .
Drop a yellow disk at column (0–6): 6 Enter
			R			
			Y	R	Y	
		R	Y	Y	Y	Y
R	Y	R	Y	R	R	R

The yellow player won