

CS5405HW (Due Thursday November 21, 2019 in class) Your Name (Uppercase)_____

This is an extension of previous homework. Turn in the complete program including (fan, start/pause, reverse, animation done in the previous HW).

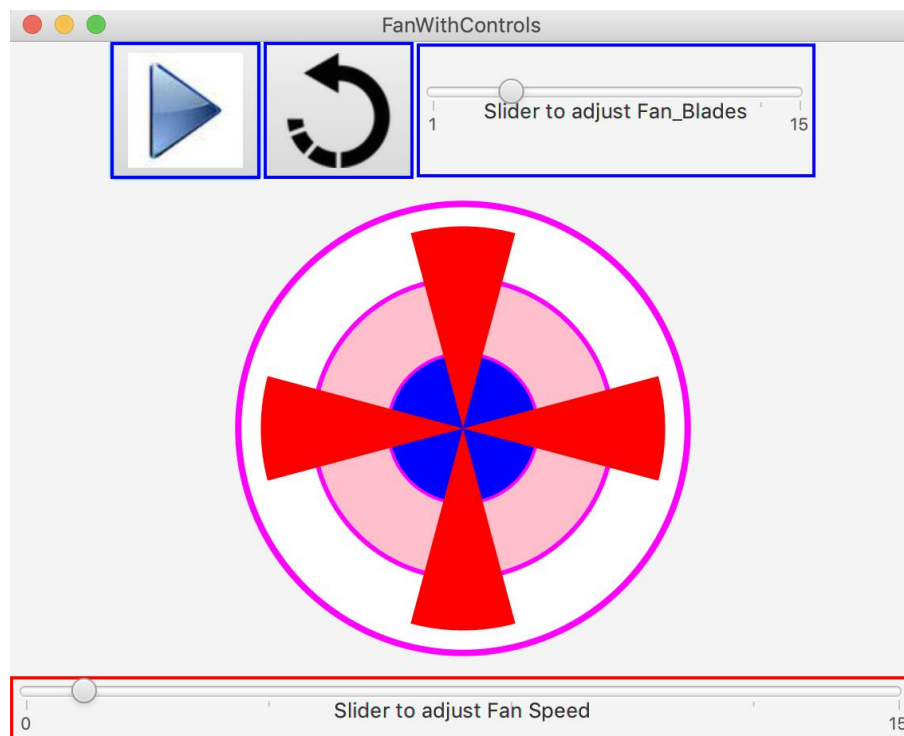
For additional interaction with animation, add

- 1. a slider to specify the number of blades in the fan,**
- 2. a slider to specify the speed of the fan.**

I showed you the demo in the class with circles and arcs. Read and understand JavaFX8 GUI Shapes.

This Homework is extension of Problem 14.9 page 587. This enforces your learning of **color and shapes circle, arc, animation Timeline**. Once you have a function for creating one fan, you can call it any number of times to create multiple fans (interactively in the next assignment). You may make your display as fancy as possible. Here is a sample of one fan. You may design the fan as you like. Ask me in the class to see fans created by your previous class fellows. You already have **user interface to start/pause, and reverse buttons**.

Demo as shown in the class (feel free to be as creative as you prefer to be)



Deliverables

1. Place .java files in source directory, compiled byte code will be in “code” directory
2. Create a jar file Demo.jar containing the folders: source and code , and a manifest file in source directory.
3. Upload Demo.jar file to Canvas
4. Bring to class a printed copy of java source code, a sample of program execution output of the program
5. Attach assignment page on top of the printed copy for grade feedback. Write your name on it.
6. You will write your own code. If any sort of plagiarism occurs, guidelines were given in the class. Copying the program will result in loss of ALL credit and will be reported to dept for proper credit.

Late Homework will not be graded. Do not ask for partial credit for turning in late, it will not happen. The grader has his own exams, classes, interviews. He is not paid for grading the same work multiple times. It is not at all fair to you or the grader to accept late work. You have a week to do it. If you have any questions ask me Tuesday. It gives you plenty of time to ask question to do it on time.

Use javadoc style comments. No UML diagram required. Use java conventions for naming classes, methods and variables. Name your program and files names accurately. You may use any java code from the examples in the book or demos on the Canvas.

Grading rubric

[10] Does it have author page with correct description: name, email, phone, ownership copyright etc.

[10] Does it have correct Problem Description

- [10] Reference, Sources are to be cited to make your work authentic and to give credit the original author(s).
- [10] does it have start, pause, and reverse buttons.
- [10] do buttons have icons instead of text labels
- [10] do start, pause, and reverse buttons function
- [10] does it have slider for number of blades, does it work?
- [10] does it have slider for fan speed, does it work?
- [10] Does it have a working correct Demo.jar file
- [10] Does Demo.jar file include NO files other than those related to program

Grading script to ensure that your program works in all aspects

```

jar -xvf Demo.jar                # extract (xvf orxf) all files from Demo.jar
mv Demo.jar DemoOld.jar         # retain Demo.jar
rm -r code                      # remove old code and docs directories, cleanup
rm -r docs
javac -d . source/*.java        # compile source code
javadoc -d docs -version -author -private -quiet source/*.java # make javadoc documentation
jar -cmf m.txt Demo.jar source code image audios # add only required folders
# execute application
java -jar Demo.jar

```

Your score out of 100 will be normalized to 10 as follows.

HW	The program doesn't have correct interaction and produces incorrect results. Does not meet naming specifications. Not turned in on time. Or Does not compile as specified.	The program had correct interaction, does not produce correct results, does not meet specifications.	The program has correct interaction, produces correct results. It does not meet all the requirements like, naming conventions, documentation.	The program has correct interaction, produces correct results. It meets all the requirements like naming conventions, documentation.
Syntax(Prog)	Inadequate 0-5	Need help 5.1-6.5	Adequate 6.6-8.5	Excellent 8.6-10
Your Score				