Lab Exercise

CCP6224 Object-Oriented Analysis And Design

Total Marks: 30% (will be scaled to 40% mathematically, since lab exercise carry 40% in the syllabus)

Due Date: 7 July 2024, 5pm (there will not be any extension of due date)

Instructions:

1. This is a group work (a group of 4 or 5 students – students must have already formed the group by now as it was already announced during lectures several times) and you must only work with among your group members. Working with anyone outside your group is considered cheating and will get zero mark. Copying from any other sources will also get zero mark. If you have trouble finding group members, please inform me by 7th June 2024. Individual work will get zero mark as this is a group exercise.

2. You will be given zero(0) marks if you do not submit on time. You are given ample time to submit earlier. Do not give excuses like internet problems etc.

3. You need to submit the following:

a. Part 1 deliverables and

b. Part 2 deliverables and

c. Part 3 deliverables

Submit only one zip folder with the file named StudentID-StudentName.zip. The zip folder should contain all the source code (.java files and any resource files), sequence diagrams, a class diagram, and a presentation video. Submission will be online and will be informed later.

4. Use only Java Swing to do the exercise. DO not use any other framework.

Question:

This group work is an extension to Lab 5 – More Swing Components and Listeners. In lab 5, we have seen how Java Swing can be used to develop a drawing program which can do free style drawing. In addition to the functionality we developed in lab 5 (i.e. free style drawing, color setting, and pen stroke size setting) you are now required to add the following functionalities.

Left- Images
-" selected image from file
- add animal, flower picture

Right
- Drawing " saved as image
- draw custom image and save the customs drawn image

1. The canvas will have two sides: 1) Left side for composing images. 2) Right side for drawing – which will be saved as an image.
2. The application should be able to add animal and flower images to the left side canvas. Both are types of creation, and in future, the application should be able to expand with more creation types.
3. The right side of the canvas allows one to draw custom image and save the custom drawn image.
4. The toolbar will have the existing text label, and icon buttons to change pen color, change pen stroke size, include animal image (from file), include flower image (from file), save custom drawn image, and include custom drawn image to the left canvas.
5. Any creation image will be able to rotate. The left canvas also can be rotated. Both canvas and creation images can be rotated independently.
6. The left canvas can be composed into another canvas (together with the images it has). The composed canvases also should be rotatable.
7. Animals, flowers, and custom images are creations. With the drag of a mouse, animal images can be flipped, flower images can be scaled, and custom images can be transposed. s a flipped image over its diagona

Since this group exercise is extending lab 5, you can refer to the solution and use the solution as the base for this work. The software design for the application must be easy to be extendable in future, and senseful.

Part 1 deliverables:

A UML class diagram and UML sequence diagrams (for each use case) for the whole program.

Part 2 deliverables:

An implementation of the whole program using Java Swing.

Part 3 deliverables:

A 10-15minute presentation video following the marking rubric sequence. Everyone in the team must participate in the presentation. Make sure the order of the presentation follows the sequence in the marking rubric.

Marking Rubric

Criteria	Poor (1 Mark)	Below Average (2 marks)	Average (3 marks)	Above Average- (4 marks)	Good (5 marks)
Class Diagram Sequence Diagram	The design is meaningless, or many parts not done, or UML notations used are wrong or presentation contains many mistakes	The diagram is drawn less senseful in the meaning it conveys or incomplete or many UML notations not correctly used. Or presentation contains many mistakes	The diagram conveys meaning but with a few parts incomplete or with some UML notations not correctly used or presentation is average.	Most of the diagram is comprehensible and illustrates most of the requirements or with some minor mistakes in the usage of UML notations or presentation has got some minor mistakes.	The diagram is comprehensible, clear, and complete. Correct UML notations used. The presentation is good. The design is senseful such that it reflects the "reality of the world".
Able to add animal images to the canvas. The class	Functionality doesn't work. Design and implementa	Some functionality does not work well. Design and	Toolbar icon works. Design and implementation are not	Toolbar icon works. Design and implementation are not	Toolbar icon works. Design and implementat

diagram design maps to the implementation. Canvas must be rotatable, and animal can be flipped.	tion are not coherent. Overall application integration fails. Presentation not convincing.	implementation are not coherent at most places. Overall the application has some flaws. Presentation has major mistakes	coherent in many places. The functionality works well. Overall application and all component integration work well. Presentation has some minor mistakes	coherent in a few places. The functionality is flawless. Overall application and all component integration work well. Presentation has some minor mistakes	ion are coherent. The functionality is flawless. Overall application and all component integration work well. The presentation is good.
Able to add flower images to the canvas. The class diagram design maps to the implementation. Canvas must be rotatable, and flower can be scaled.					
Able to create and add custom images to the canvas. The class diagram design maps					

to the implementation. Canvas must be rotatable, and custom image can be transposed.					
Able to add animal, flower, and custom images to the canvas. The canvases can be composed to one another. The class diagram design maps to the implementation. Each canvas must be rotatable, and each image can be either flipped,					

scaled, or transposed depending on their type.					