Metropolitan State University, Saint Paul, Minnesota

ICS 372 Object-Oriented Design and Implementation Exam 3 Part 2

**Date and Time** 6:00 PM (or a few minutes later) on April 29, 2021

**Duration**: 3 hours (maximum)

**Points** 80

The exam contains two questions requiring you to write Java code; there is also a question that requires you to create the state transition table for an FSM.

Then there is a set of multiple-choice/true-false/multi-select/short-answer questions for a total of 20 points. That is under Quizzes (under Assessments).

**Ground Rules**

1. This is an open book/open notes test. You may use any resources on the internet. The only restriction is that you are not allowed to communicate in any way with anyone: like phone call, online chat, email, posing questions on online forums, etc. So, this is an individual effort.
2. You must have your video on. Mute the audio. If you need to ask me a question, please send a message privately on Zoom chat. I may move you to a waiting room and we can talk briefly.
3. If you have to leave the exam for more than a couple of minutes, please get permission from your instructor. Use the approach in (2) above.
4. Two of the questions ask you to write Java code. These must be submitted as **Eclipse Java** project(s) – you may have both projects in the same Eclipse project - to the dropbox for Exam 3. Ensure that any Java code you write is syntactically correct and works as specified. Name the projects <Your-last-name>Exam3Q1 and <Your-last-name>Exam3Q2.
5. Submit the answer to the state transition table as a Word file or a PDF file. Do not submit Excel spreadsheets or picture files. There is penalty if you submit anything other than a Word or PDF file. Upload the file to the dropbox.
6. Please follow the directions carefully, so you don’t lose credit.
7. The exam must be turned in on time. I will announce the deadlines at the start of exam. If you are late starting the exam, this may mean a loss of time for you to take the exam.
8. Late submission will incur a penalty of 1 point per minute. No submission will be accepted after 9:20 PM.

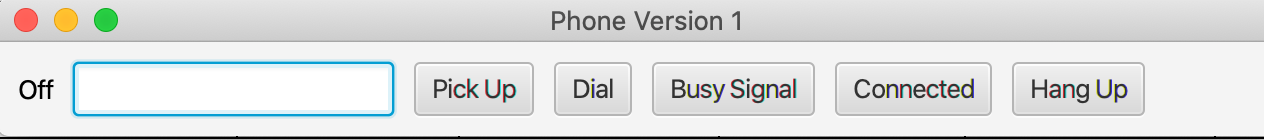
**Question 1** (30 points)

The following state transition table was created for a phone.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Pick Up | Dialing Event | Hang Up | Busy Signal | Connected Signal |
| Off | Dial Tone | Off | Off | Off | Off |
| Dial Tone | Dial Tone | Dialing | Off | Dial Tone | Dial Tone |
| Dialing | Dialing | Dialing | Off | Busy | Connected |
| Busy | Dial Tone | Busy | Off | Busy | Busy |
| Connected | Connected | Connected | Off | Connected | Connected |

A partial implementation is given in the Java project Exam 3 Q1. Complete the implementation, so it works as in the above table. You must “continue” the implementation, rather than do a complete redesign and reimplementation. The naming conventions, packages, and style must match the existing implementation.

When the program starts, the GUI appears as below.



1. The status of the phone is shown in the Text widget (currently shown as Off).
2. The user enters the phone number period in the Textfield. (Blank in the picture.)
3. The user presses the Pickup button to simulate picking up of the phone.
4. The Dial button simulates dialing. The phone number is immaterial in this program.
5. The call can be canceled by pressing Hang Up.
6. The Busy Signal and Connected buttons simulate the busy and connected signals that come to the phone after a call is placed.

The state transition table is not correctly implemented. Implement it, without modifying the rest of the program.

The user interface must appear and function as given in the above picture and the attached video.

No documentation is needed. But other coding standards (naming, readability, etc.) apply.

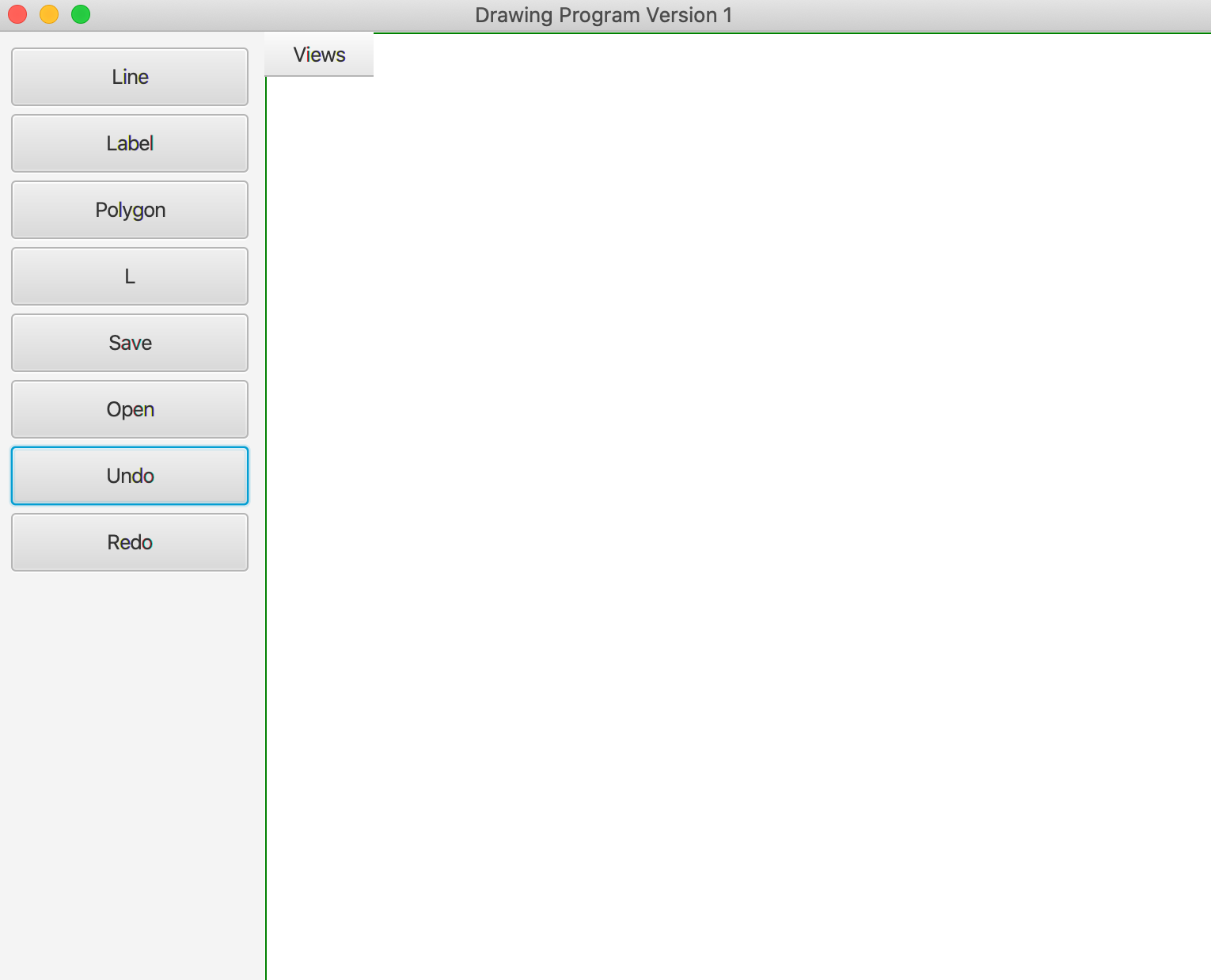
1. Rename the project as <Your-last-name>Exam3Q1. Make sure it is an Eclipse Java project with no modules. You **will** lose 1 or 2 points if you don’t do this. So take care of it right away. Improper format and names are a problem for me to import and/or locate your project.
2. If you add methods to existing classes, put them at the end of the classes to which they belong, so they are easy to locate. There **will** be penalty for not following this requirement.

**Question 2** (30 points)

In the drawing program implementation (Exam 3 Q2), add code to implement the functionality to draw an L shape. The L shape could be upside down, upright, and laterally inverted.

The user creates an L by pressing the L button and then clicking two points. See the behavior in the uploaded video file.

The interface must appear and function as given in the following picture and the attached video. This includes the order of the buttons.



Your implementation must not depend in any way on the shapes already supported in the system. The implementation must conform to the style and approach employed for the other shapes.

Remember that the program needs only two point clicks. The L could be upright or upside down. The first point specifies the coordinates of the free end of the vertical line of the L.The second point (in conjunction with the first) specifies the coordinates for everything else.

No documentation is needed. But other coding standards (naming, readability, etc.) apply.

1. Rename the project as <Your-last-name>Exam3Q2. Make sure it is an Eclipse Java project with no modules. You **will** lose 1 or 2 points if you don’t do this. So take care of it right away. Improper format and names are a problem for me to import and/or locate your project.
2. If you add methods to existing classes, put them at the end of the classes to which they belong, so they are easy to locate. There **will** be penalty for not following this requirement.

**Question 3.** (20 points) Analyze the following problem and use FSM modeling to arrive at the state transition table. All states and events must be clearly named. If you use improper names or numbers for the states or use unconventional ways of representing state transitions, I will not bother to interpret your answers and the credit will be 0 for this question. The table should be created using Word. Tables that are handwritten or have badly aligned entries will not be given credit.

Consider a coffee maker that is composed of the following:

1. It has three buttons: on, off, warm.
2. It has a water tank into which water can be poured.
3. There is a place to put coffee. Let us call this coffee holder.
4. There is a coffee pot, which collects coffee made by the coffee maker.
5. Underneath the coffee pot, there is a plate, which can be heated so the coffee can be kept warm, if needed.

It works as below.

1. The coffee maker is able to sense whether or not there is water in the water tank. For example, if the tank is empty and we pour water into the tank, the coffee maker immediately senses that there is water.
2. When there is water in the coffee maker, if the on switch is pressed, the coffee maker senses the water, and boils and percolates it through the coffee holder and thus makes coffee. (We are assuming there is coffee powder in the coffee holder.)
3. Pressing the on button when the machine is percolating coffee has no extra effect. Otherwise, if there is water when the button is pressed, coffee will be made, no matter what the coffee maker was doing when the on button was pressed.
4. If the coffee maker senses no water, it does not attempt to percolate the water through the coffee holder.
5. When the water is completely gone, a sensor sends a “no water” signal to the coffee maker.
6. After receiving the “no water” signal, the coffee maker will not attempt to boil the water or percolate it through the coffee powder.
7. As the coffee is being made, the process can be stopped by pressing the off button. The machine turns off, but is ready to make more coffee, if there is water.
8. If the coffee making process ends because of the “no water” signal, the machine automatically keeps the coffee warm for one hour.
9. The warming process ends and the machine turns off either when the off button is pressed or after one hour, whichever occurs first.
10. Pressing the warm button when the coffee maker is already warming the coffee adds one hour to the warming process. Pressing the same button when the coffee is being made does not have any effect.
11. Pressing the warm button when the machine is off, starts the warming process, which lasts one hour.
12. Pressing the off button always turns the machine off.