What is a GUI?

- GUI is a Graphical User Interface that is a program that consists of code that runs from the top down. This is especially useful for users to see the visual display of the code. Which consists of labels, buttons and other things for the user to use just like in a game, software or application of some sort.

Explain how code is executed in an event-driven application (GUI).

- In the java Gui application using Swing code, its function is to react to the user's interaction with a component called the event function. Which makes programmers the ability to write code to do something right after the event. This is how an event-driven application works.
- The code runs like a flow of program execution and is determined by the events established by the programmer. Usually, the user acts, like a mouse clicking or a key is pressed or a message from the computer systems. Then something appears on the screen or does something.

Can components be added directly to a frame? Explain.

- When components are added directly to the frame, the program automatically adds the component to the j-panel, making it kinda adding to each other while editing. The programmer has to make correct adjustments to the components to make them fit, adjustable or fit inside the code in order. If not, I might not directly end up in the frame.

Can a label respond to events? Explain.

- Especially, J-labels can respond to events because the labels can have a method where the programmer can insert code for it to do something, like displaying text and images, if-else statements, MouseListener requires a Mouse adapter. Overall the label consists of a label name, and other functions as the frame, text field, button, etc.

Why do you think a GUI needs to be run from an event dispatching thread?

- The GUI needs to be run from an event dispacing thread because without the thread, most Swing object methods are not “thread-safe”. Making multiple threads risk of interference or memory consistency issues and errors during the time. Sometimes the Java Program will tell the programmer whether the Swing thread is safe or not.

What are the differences between a label and a button?

- J-Label is a label that can display, hold or release certain text/String or image sources of a user when they press a button. While the button is like the confirm button that activates the code to release the information the J-label is holding. The button is also like an activator or a light switch, that lights the code to life when the user activates it. Also, the J-label can be edited by the user, meaning the user can overwrite text onto the box while using the program unless the programmers shut it off.
-