Question Answering Bonus

1. What is Event driven programming ?

   Event-driven programming is a programming paradigm where the flow of execution of a program is determined by events—specific actions or occurrences that the program listens for and responds to. These events can be user actions like mouse clicks or keyboard inputs, system-generated events like messages from other programs or sensor outputs, or other asynchronous occurrences. Instead of following a linear or sequential flow, event-driven programs wait for events and execute event handlers or callback functions when corresponding events occur.

   In this model, key components include event sources (which generate events), event listeners (which detect events), and event handlers (which execute the response code). The program flow is thus controlled by events, making applications highly responsive, interactive, and scalable. This paradigm is commonly used in graphical user interfaces, real-time systems, and distributed architectures where asynchronous processing and decoupled components are critical.

   The event-driven approach improves application responsiveness by reacting to events as they happen, supports loose coupling between components, and enhances scalability by allowing independent event producers and consumers to work asynchronously. Overall, event-driven programming facilitates writing modular, maintainable, and dynamic applications that react flexibly to inputs and system changes.