1. What is a finite state machine? What are the different types of finite-state machines?

   A Finite State Machine (FSM) is a computational model used to design algorithms and systems with a limited number of states. It consists of a finite set of states, transitions between those states, and actions based on state changes.

   There are 2 types of FSM: Mealy and Moore.
2. Whats is the difference between Mealy and Moore machine?

   Mealy machine
   - In Mealy machine the outputs depend on both the current state and the current input. The output is generated on transitions.
   - More complex to design.

   Moore machine
   - In Moore machine the outputs depend only on the current state, not on the input. The output is associated with states rather than transitions.
   - More simplier to design.
3. What are the applications and limitations of FSM?

   Applications
   - Used in parsing algorithms, regular expression matching, and designing algorithms for string processing.
   - Employed in designing sequential logic circuits, like counters and registers.
   - They are also used to detect a squence in a given input set.

   Limitations
   - They have limited memory as they only store the current state.
   - They have fixed number of states, this means the structure is rigid, which makes modifying or reusing the FSM harder.
   - It is highly difficult to scale.
4. How do you handle exceptions or error states in finite state machine?
   - We can add extra states to be triggered incase of an error.
   - If state machine detects an internal error during a state transition logic it may throw an exception. Before this exception is processed internally, user is given a chance to intercept.

5. Explain the below concepts in FSM
   a. Trigger

      A trigger refers to an event or condition that causes a transition from one state to another. Triggers are fundamental in determining how and when state transitions occur based on the input or external conditions.

   b. Transition

      A transition is when the state machine changes its state from current state to a different one.

   c. State diagram

      State diagram is a graphicaal representation of a State machine. This helps write the code for any state machine.

   d. Super state

      Super state is a higher-level abstraction used to group together related states and manage their transitions more efficiently.

   e. Guard conditions

      Guard conditions are logical expressions or conditions that must be true for a transition between states to occur. They act as constraints or checks that govern whether a transition should be executed based on the current state and input or context.

f. Deadlock

   Deadlock refers to a situation where the FSM becomes stuck in a state or a set of states from which it cannot transition to any other state. This often occurs because there are no valid transitions available based on the current state and input conditions this stops the system.

g. Determinisim

   Determinism refers the to beahiour of the FSM to be predictable.

h. Epsilon transitions

   An epsilon transition is a transition in an Non-deterministic FSM that allows the automaton to move from one state to another without consuming any input symbol.