***Module 1: Introduction to System Software***

* + *1. Compare System Software and Application Software with examples.*
  + *- (2024 Q1A, 2023 Q1a, 2019 Q1a, May 2019 Q1a)*
  + *2. Define System Programming. Explain the goals of system software.*
  + *- (2023 Q1a, 2019 Q1a)*

***Module 2: Assemblers***

* + *1. Explain the forward reference problem in assemblers. How is it handled in a single-pass assembler?*
  + *- (2024 Q1C, 2023 Q1b, May 2023 Q1D, May 2019 Q1c)*
  + *2. Draw and explain the flowchart of Pass-I of a two-pass assembler. Mention the databases used.*
  + *- (2024 Q2A, 2023 Q2a, May 2019 Q5a, Dec 2019 Q5a)*
  + *3. Explain the following assembler directives with examples: MOT, POT, ST, BT.*
  + *- (May 2019 Q2c)*
  + ***Module 3: Macros and Macro Processor***
  + *1. Explain conditional macro expansion with a suitable example.*
  + *- (2024 Q6B, 2023 Q1c, May 2019 Q1d)*
  + *2. Explain the design of a two-pass macro processor with a flowchart.*
  + *- 2024 Q4A, May 2023 Q3B)*
  + *3. What are advanced macro facilities? Discuss nested macros with an example.*
  + *- (2024 Q6B, May 2019 Q5b)*
  + ***Module 4: Loaders and Linkers***
  + *1. What is a loader? Explain the functions of a loader in detail.*
  + *- (2024 Q1D, 2023 Q1d, Dec 2019 Q1b, May 2019 Q1b)*
  + *2. Explain the working of a Direct Linking Loader with a diagram.*
  + *- (2024 Q3A, May 2023 Q5B, Dec 2019 Q4c)*
  + *3. What is dynamic linking? Explain Dynamic Linking Loader in detail.*
  + *- (2023 Q6a, Dec 2019 Q4c)*
  + ***Module 5: Compilers (Analysis Phase)***
  + *1. Explain the phases of a compiler with a suitable example (e.g., a = b \* c + 10).*
  + *- (2024 Q6A, 2023 Q5b, Dec 2019 Q2b, May 2019 Q2b)*
  + *2. Compute FIRST and FOLLOW for the given grammar:*
  + *S → Aa*
  + *A → BD*
  + *B → b | ε*
  + *D → d | ε*
  + *- (2023 Q1d, Dec 2019 Q2a, May 2019 Q2a)*
  + *3. Construct an SLR/LR(0) parsing table for the grammar:*
  + *S → (S)S | a*
  + *Parse the input string ( )( ).*
  + *- (May 2023 Q2A, Dec 2019 Q6b-iii)*
  + ***Module 6: Compilers (Synthesis Phase***
  + *1. Generate three-address code for the following code snippet:*
  + *while (a < b) do*
  + *if (c < d) then x = y + z;*
  + *else x = y - z;*
  + *- (2024 Q2B, May 2023 Q1A, May 2019 Q2b)*
  + *2. Explain machine-independent code optimization techniques with examples:*
  + *- Common subexpression elimination*
  + *- Dead code elimination*
  + *- Constant propagation*
  + *- (2024 Q4B, 2023 Q6b, Dec 2019 Q3b, May 2019 Q3b)*
  + *3. What are basic blocks and flow graphs? Construct them for a given three-address code.*
  + *- n(May 2023 Q3A, 2023 Q5a)*
  + ***Top 10 Full Questions with Highest Recurrence***
  + *1. Two-pass assembler design flowchart and explanation.(4+ times)*
  + *2. Compare system software vs. application software.(4 times)*
  + *3. Phases of a compiler with example.(4 times)*
  + *4. Functions/types of loaders (Direct Linking, Dynamic Linking). (4 times)*
  + *5. Conditional macro expansion with example(3 times)*
  + *6. FIRST & FOLLOW computation for a grammar.(3 times)*
  + *7. Three-address code generation for loops/conditionals.\*\* \*(3 times*
  + *8. Code optimization techniques (machine-independent).\*\* \*(3 times)\**
  + *9. Forward reference problem in assemblers.\*\* \*(3 times)\**
  + *10. SLR/LR(0) parsing table construction. (2 times)*