

Importance

Module-1(System Software)

- What is System Software and application software?
- Differentiate between them (minimum 10 points)
- Indicate the order in which following system programs are used from developing program up to its execution: Assembler, Loader, Linker, Macroprocessor, Compiler, Editor.

Module-2(Assembler)

- Explain with the help of neat flowchart the working of a two-pass assembler. Clearly showing the organization and uses of the databases.(MUST DO)
- With reference to assembler explain the following tables with suitable example. POT, MOT, ST, LT, BT

Module-3(Macros and Macro Processor)

- Explain different pseudo-ops used for conditional macro expansion along with example(AIF and AGO with example).
- Explain with neat and labelled flowchart the working of Multi-pass(Two pass) Macroprocessor. (MUST DO).

Module-4(Loaders and Linkers)

- What is loader and explain the Function of Loader with example? (MUST DO)
- Explain design and flowchart of absolute Loader?(MUST DO)
- Explain the working of Direct Linking Loader with example show entry in different built by DLL?
- Difference between Dynamic Loading and Dynamic Linking with example?(EASY)

Module-5(Editor)

- Explain design of an editor along with neat diagram?
- Explain types of Editor.



Module-6(Compiler)

- What are the different phases of compiler? Illustrate compilers internal presentation of source program for the following statement after each phase.. $position = initial + rate * 60$ (MUST DO)
- Compare the performance of compilers and interpreters.(EASY)

Module-7(Lexical Analyser)

- Write short note on L.E.X(YACC is also questioned with it).
- Explain role of lexical analyser.

Module-8(Syntax Analyser)

- Error Recovery techniques used in compiler.
- Distinguish LL parser and LR parser.
- Explain process of elimination of Left Recursion and Left Factoring.
- Explain left recursion present in following grammar(Remove Direct and Indirect recursion both)

$S \rightarrow Aa \mid b$

$A \rightarrow Ac \mid Sd \mid \Sigma$

- Top down and bottom up + Recursive descent parser.
- Find First and follow.
- Predictive parser(LL1).
 - a. left recursion eliminate.
 - b. first and follow.
 - c. parsing table.
 - d. parse the string
- Operator Precedence Parser. (LR0, SLR1, CLR1 and LALR1)

Module-9(Syntax Directed Translation)

- Explain synthesized and Inherited attributes used in syntax directed definitions?(EASY)
- Explain Syntax Directed Translation. Give syntax directed definition to translate infix expression to postfix expression?(EASY)



Module-10(Intermediate Code Generation)

- Explain three-address code, its types and also implementation of three address statements.(MUST DO)
- Explain Backpatching with an example.(EASY)
- Explain DAG.
- Draw Syntax tree and directed Acyclic Graph(DAG) for expression $(a*b)+(c-d)-(a*b)+b$

Module-11(Code Generation)

- What is flow graph? State its significance in code generation.
- What are different issues in code generation, explain in detail.

Module-12(Code Optimization)

- What is code optimization? What are various strategies for code optimization?(MUST DO)
- Explaining Peephole optimization along with an example.(MUST DO)
- Explain loop optimization with example.(EASY)

Module-13(Run Time Storage)

- Explain Run Time Organization in Details.
- What is activation record? Draw the diagram of General Activation record and explain the purpose of different fields of an activation record.(MUST DO)
- What is binding? Explain the static and dynamic binding.(EASY)

Module-14

- YACC
- Java Compiler Environment.(MUST DO)

Do share with your friends and help them :)