SP Important topics

Date

Chapter 1 :- Introduction

→ Problem Oriented Vs Procedure oriented Languages.
→ User Centric View Vs. System-Centric View of System Software.
→ Explain levels of System Software.
→ What is overlay Structured Program.
→ Explain types of System Software.
→ System Software Vs Application Software.

Page No.

Date

Chapter 2 Language Proc...

→ Definitions of types of Chaps
→ Types of Grammars ||||
→ Regular expression generation examples
→ Life Cycle of Source Program ||
→ Explain language Processing activities
→ Explain Phases of language Processor ||
→ Explain lexical analyzer of language Processor ||
→ Explain main task of Semantic analysis Phase
→ What is ambiguous grammar? Explain with example
→ Which activity reduce Specification Gap?
→ Define Simple Phase grammar
→ Explain L-attributed definition ||

Chapter 3- Assemblers

Date

* → Life Cycle of Source Program ||
→ Generate tables from Given Assembly Code ||||
→ Explain how forward references are Solved & Explain back Patching.
→ What is Symbol table? Explain how One Can organize Symbol table using linear data Structure.
→ What are advanced assembler directives. (Explain any two.) ||
→ An assembly Programs Contains the Statement.

X EQU Y+25

Indicate how equ will be Processed if (i) Y is a back reference (ii) Y is a forward reference.
→ Types of intermediate Code Representation ||||
→ Explain two Pass assembler
→ Draw flowchart of Maintaining Table of Incomplete Instruction (TII)
→ Explain Single Pass assembler
→ Literal Vs. Constant
→ Explain EQU & Origin

||

Date

Chapter 4 Macro

→ Explain advance Macro Programming facilities.

→ Explain use and field of following tables of Macro KPDTAB, MDT, EUTAB, SSTAB.

→ Explain following expansion time loop with Example (i) REPT Statement (ii) IRP Statement ||

→ Define (i) Expansion time Variable (ii) Semantic Expansion (iii) Positional Parameter (iv) Maro Pre-Processor.

→ Draw & Explain flowchart of 1 Pass macro Processor.

→ Basic tasks of Macro PreProcessor

→ Explain attributes of formal Parameters of Macro

→ Use of Stack in Nested Macro Calls.

→ Macro Prosse PreProcessor vs Macro Assembler.

→ Brief design of Macro assembler

→ Give Example of Nested Macro Call with data Structure.

→ Write a Macro Program with Condition Expansion or Explain expansion time loops

→ Explain attributes of formal Parameters & Expansion time Variables ||

→ Explain Positional Parameter & Expansion time Variable.

Date

Chapter 5 Linkers & loaders

→ What is Program relocation ? And how it is Performed by Linker ? Explain with example. ||

→ Explain Absolute loader with example. ||||

→ Explain how relocation is Performed by linker. |||

→ What is overlay ? Explain linking of overlay Structured Program

→ What is Object Module

→ What is bootstrap loader.

→ Explain in brief design the Linker.

→ Explain Self relocating Programs. ||

→ Definitions of types of addresses.

→ Explain the term loader with its basic functions.

Date

Chapter 6 Scanning & Parsing

→ Construct N.F.A & D.F.A for Given Regular expression. [(0|1)* 00 1 #] ||

→ Explain recursive descent Parsing algorithm. ||

→ Explain left recursion & left factoring ||

→ LL(1) Parser example ||

→ Develop regular expression & DFA for
(i) a real number with optional integer & fractional part.
(ii) a Comment String in the C++ language

→ Create / Operator Precedence table for ... (type of examples) [Winter 2016 Q-3 c] ||

→ Remove left recursion.

→ Construct Optimized DFA 0* 1* (0|1) #

→ Write an algorithm for Practical approach of top down Parsing.

→ Predictive Parsing Example [Winter 2017 Q-2 c]

→ Construct an Operator Precedence Matrix for the Operators of a grammar Containing Arithmatic, Relational & Boolean Operators.

→ Explain naive bottom-up Parsing algorithm.

Date

Chapter 7: Compilers

→ List out and explain various optimizing transformations performed by Compiler

→ Given the following expression $x = -a * b + -a * b$.
(i) Write three address codes.
(ii) if Possible optimize 3 address code.
(iii) Give triple implementation of 3 address code.

→ How Compiler implements Scope rules?

→ Explain use of Static Pointer & dynamic Pointer in Compiler

→ What is Peephole Optimization?

→ State Different Storage allocation techniques Explain Static & Stack allocation.

→ Generate Quadruple, triple & Indirect triple for following expression. ans = a + b * c / z - o.

→ Data Structure used in automatic dynamic Memory allocation

→ Explain use of Value No. in local optimization.

→ Explain front-end of toy Compiler

→ Expression tree Example

→ Explain Operand descriptor & Register descriptor. for a*b.

→ Explain Memory allocation in blocked Structure.

→ Explain triple & Quadruple Representations.

→ Explain any 3 code optimization techniques.

→ Explain term binding and binding times.

Chapter 8 Interpreters

→ What is interpreter ? Explain benefits of interpreter
→ Compiler vs. Interpreter
→ What is interpreter ? Explain pure & impure interpreters |||
→ ~~What is absolute loader~~
→ Explain drawbacks & benefits of Interpretation

||

Chapter 9 → Software tools

→ Explain design of an editor