

MCA-14-21**SYSTEM PROGRAMMING****Maximum marks: 100 (External: 80, Internal: 20)****Time: 3 hours**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

UNIT - I

System Software: Definition, Goals of System Software, Program Development and Production Environments, Software Portability, Programs as components, Quick and Dirty Programming, User-Centric and System-Centric view of System Software.

Language Processors: Types of Language Processors, Program Generation, Program Execution, Program Translation and Interpretation, Fundamentals of Language Processing, Symbol Tables.

UNIT - II

Assemblers: Elements of Assembly language Programming, Pass Structure of Assemblers, Design of Two-pass assembler, Intermediate code forms, Program Listing and Error reporting, Organizational and Design issues in assemblers.

Macros and Macro Preprocessors: Macro Definition and Call, Macro expansion, Nested Macro calls, Design of a Macro preprocessor, Processing of Macro definitions, Use of Stack in expansion of macro calls, Design of a macro assembler

UNIT - III

Linkers and Loaders: Linking & Relocation, Design of a Linker, Self-Relocating, Dynamic Linking, Linking for program overlays, Loaders, Absolute and Relocating loaders.

Scanning and Parsing: Chomsky hierarchy of formal languages, Ambiguous grammars, Scanning, Parsing: Top-down and Bottom-up Parsing.

UNIT - IV

Compilers and Interpreters: Binding and Binding times, Data Structures of compilers, Scoping rules, Memory allocation, Static and dynamic memory allocation and deallocation, Recursion, Compilation of expressions, Postfix notations, Expression trees, Compilation of Control structures, Code Optimization, Local and Global optimization, Overview and benefits of interpretation, Pure and impure interpreters.

Text books:

1. Dhamdhare D.M, "System programming", Tata McGraw-Hill.
2. Beck L. Leland, "System Software", Pearson Education.

Reference Books:

1. Aho, Sethi, & Ullman, "Compilers Principles, Techniques and Tools", Pearson Education.
2. Donovan J. John, "System Programming", Tata McGraw Hill.