

1. What is a compiler?

- a) A hardware device
- b) A software program
- c) A peripheral device
- d) A storage device

Answer: b) A software program

Explanation: A compiler is a software program that translates source code written in a high-level programming language into machine code that can be understood and executed by a computer.

2. Which of the following is a major data structure used in compilers?

- a) Arrays
- b) Stacks
- c) Trees
- d) Linked lists

Answer: c) Trees

Explanation: Trees, particularly Abstract Syntax Trees (ASTs), are commonly used in compilers for representing the structure of source code.

3. How many types of compilers are there based on the translation process?

- a) One
- b) Two
- c) Three
- d) Four

Answer: b) Two

Explanation: Compilers can be classified into two types based on the translation process: Single-pass compilers and Multi-pass compilers.

4. Which part of the compiler translates high-level code into intermediate code?

- a) Front-end

- b) Back-end
- c) Middle-end
- d) Preprocessor

Answer: a) Front-end

Explanation: The front-end of the compiler translates high-level source code into an intermediate representation or code.

5. Which part of the compiler generates machine code from the intermediate code?

- a) Front-end
- b) Back-end
- c) Middle-end
- d) Postprocessor

Answer: b) Back-end

Explanation: The back-end of the compiler generates machine code from the intermediate representation produced by the front-end.

6. Which model of compilation involves two phases: analysis and synthesis?

- a) Single-pass model
- b) Multi-pass model
- c) Analysis-synthesis model
- d) Interpretation model

Answer: c) Analysis-synthesis model

Explanation: The analysis-synthesis model involves two main phases: analysis, where the source code is analyzed, and synthesis, where the target code is generated.

7. Which phase of the compiler checks the syntax of the source code?

- a) Lexical analysis
- b) Semantic analysis
- c) Syntax analysis
- d) Code generation

Answer: c) Syntax analysis

Explanation: Syntax analysis, also known as parsing, checks the syntax of the source code to ensure it follows the rules of the programming language's grammar.

8. What is the purpose of lexical analysis in a compiler?

- a) Checking syntax errors
- b) Generating machine code
- c) Tokenizing the source code
- d) Optimizing code

Answer: c) Tokenizing the source code

Explanation: Lexical analysis breaks the source code into tokens such as keywords, identifiers, operators, etc., which are the basic building blocks of the language.

9. Which phase of lexical analysis involves grouping characters into tokens?

- a) Input buffering
- b) Token specification
- c) Token recognition
- d) Token generation

Answer: c) Token recognition

Explanation: Token recognition involves identifying and grouping characters into tokens based on predefined patterns.

10. Which tool is used for generating lexical analyzers automatically?

- a) Yacc
- b) Bison
- c) Flex
- d) ANTLR

Answer: c) Flex

Explanation: Flex is a tool for generating lexical analyzers automatically based on regular expressions specified by the user.

11. What does LEX stand for?

- a) Lexical Extraction
- b) Lexical Extension
- c) Lexical Analyzer Generator
- d) Lexical Expression

Answer: c) Lexical Analyzer Generator

Explanation: LEX is a lexical analyzer generator that produces lexical analyzers based on user-specified regular expressions.

12. Which phase of the compiler handles input buffering?

- a) Lexical analysis
- b) Syntax analysis
- c) Semantic analysis
- d) Code generation

Answer: a) Lexical analysis

Explanation: Input buffering involves reading characters from the source file and buffering them for processing by the lexical analyzer.

13. Which data structure is commonly used for implementing symbol tables in compilers?

- a) Arrays
- b) Linked lists
- c) Hash tables
- d) Trees

Answer: c) Hash tables

Explanation: Hash tables are commonly used for implementing symbol tables due to their efficiency in searching, insertion, and deletion operations.

14. Which phase of the compiler checks the meaning of the source code?

- a) Lexical analysis
- b) Syntax analysis

- c) Semantic analysis
- d) Code generation

Answer: c) Semantic analysis

Explanation: Semantic analysis checks the meaning of the source code, ensuring that it adheres to the language's semantic rules.

15. Which phase of the compiler translates intermediate code into machine code?

- a) Lexical analysis
- b) Syntax analysis
- c) Semantic analysis
- d) Code generation

Answer: d) Code generation

Explanation: Code generation translates the intermediate representation produced by earlier phases into machine code for execution by the target machine.

16. Which part of the compiler optimizes the generated code for efficiency?

- a) Lexical analysis
- b) Syntax analysis
- c) Semantic analysis
- d) Code optimization

Answer: d) Code optimization

Explanation: Code optimization improves the efficiency of the generated code by applying various transformations to reduce execution time or memory usage.

17. Which phase of the compiler resolves references to variables and functions?

- a) Lexical analysis
- b) Syntax analysis
- c) Semantic analysis
- d) Code generation

Answer: c) Semantic analysis

Explanation: Semantic analysis resolves references to variables and functions by checking their declarations and ensuring their correct usage.

18. Which part of the compiler translates source code into an intermediate representation?

- a) Front-end
- b) Back-end
- c) Middle-end
- d) Preprocessor

Answer: a) Front-end

Explanation: The front-end of the compiler translates source code into an intermediate representation, which is then processed by the back-end.

19. Which phase of the compiler removes unnecessary code and reduces code size?

- a) Lexical analysis
- b) Syntax analysis
- c) Semantic analysis
- d) Code optimization

Answer: d) Code optimization

Explanation: Code optimization removes unnecessary code and reduces code size to improve performance and reduce memory usage.

20. Which part of the compiler performs error checking and reporting?

- a) Front-end
- b) Back-end
- c) Middle-end
- d) Preprocessor

Answer: a) Front-end

Explanation: The front-end of the compiler performs error checking and reporting to identify syntax and semantic errors in the source code.

Related posts:

1. Syntax Analysis & Syntax Directed Translation MCQs
2. Type Checking & Run Time Environment MCQs
3. Code Generation MCQs
4. Code Optimization MCQs
5. Introduction to Energy Science MCQ
6. Ecosystems MCQ
7. Biodiversity and its conservation MCQ
8. Environmental Pollution mcq
9. Social Issues and the Environment MCQ
10. Field work mcq
11. Discrete Structure MCQ
12. Set Theory, Relation, and Function MCQ
13. Propositional Logic and Finite State Machines MCQ
14. Graph Theory and Combinatorics MCQ
15. Relational algebra, Functions and graph theory MCQ
16. Data Structure MCQ
17. Stacks MCQ
18. TREE MCQ
19. Graphs MCQ
20. Sorting MCQ
21. Digital Systems MCQ
22. Combinational Logic MCQ
23. Sequential logic MCQ
24. Analog/Digital Conversion, Logic Gates, Multivibrators, and IC 555 MCQ
25. Introduction to Digital Communication MCQ

26. Introduction to Object Oriented Thinking & Object Oriented Programming MCQ
27. Encapsulation and Data Abstraction MCQ
28. MCQ
29. Relationships – Inheritance MCQ
30. Polymorphism MCQ
31. Library Management System MCQ
32. Numerical Methods MCQ
33. Transform Calculus MCQ
34. Concept of Probability MCQ
35. Algorithms, Designing MCQ
36. Study of Greedy strategy MCQ
37. Concept of dynamic programming MCQ
38. Algorithmic Problem MCQ
39. Trees, Graphs, and NP-Completeness MCQ
40. The Software Product and Software Process MCQ
41. Software Design MCQ
42. Software Analysis and Testing MCQ
43. Software Maintenance & Software Project Measurement MCQ
44. Computer Architecture, Design, and Memory Technologies MCQ
45. Basic Structure of Computer MCQ
46. Computer Arithmetic MCQ
47. I/O Organization MCQ
48. Memory Organization MCQ
49. Multiprocessors MCQ
50. Introduction to Operating Systems MCQ
51. File Systems MCQ
52. CPU Scheduling MCQ

- 53. Memory Management MCQ
- 54. Input / Output MCQ
- 55. Operating Systems and Concurrency
- 56. Software Development and Architecture MCQ
- 57. Software architecture models MCQ
- 58. Software architecture implementation technologies MCQ
- 59. Software Architecture analysis and design MCQ
- 60. Software Architecture documentation MCQ
- 61. Introduction to Computational Intelligence MCQ
- 62. Fuzzy Systems MCQ
- 63. Genetic Algorithms MCQ
- 64. Rough Set Theory MCQ
- 65. Introduction to Swarm Intelligence, Swarm Intelligence Techniques MCQ
- 66. Neural Network History and Architectures MCQ
- 67. Autoencoder MCQ
- 68. Deep Learning MCQs
- 69. RL & Bandit Algorithms MCQs
- 70. RL Techniques MCQs
- 71. Review of traditional networks MCQ
- 72. Study of traditional routing and transport MCQ
- 73. Wireless LAN MCQ
- 74. Mobile transport layer MCQ
- 75. Big Data MCQ
- 76. Hadoop and Related Concepts MCQ
- 77. Hive, Pig, and ETL Processing MCQ
- 78. NoSQL MCQs Concepts, Variations, and MongoDB
- 79. Mining social Network Graphs MCQ

- 80. Mathematical Background for Cryptography MCQ
- 81. Cryptography MCQ
- 82. Cryptographic MCQs
- 83. Information Security MCQ
- 84. Cryptography and Information Security Tools MCQ
- 85. Data Warehousing MCQ
- 86. OLAP Systems MCQ
- 87. Introduction to Data & Data Mining MCQ
- 88. Supervised Learning MCQ
- 89. Clustering & Association Rule mining MCQ
- 90. Fundamentals of Agile Process MCQ
- 91. Agile Projects MCQs
- 92. Introduction to Scrum MCQs
- 93. Introduction to Extreme Programming (XP) MCQs
- 94. Agile Software Design and Development MCQs
- 95. Machine Learning Fundamentals MCQs
- 96. Neural Network MCQs
- 97. CNNs MCQ
- 98. Reinforcement Learning and Sequential Models MCQs
- 99. Machine Learning in ImageNet Competition mcq
- 100. Computer Network MCQ
- 101. Data Link Layer MCQ
- 102. MAC Sub layer MCQ
- 103. Network Layer MCQ
- 104. Transport Layer MCQ
- 105. Raster Scan Displays MCQs
- 106. 3-D Transformations MCQs

- 107. Visualization MCQ
- 108. Multimedia MCQs
- 109. INTRODUCTION Knowledge Management MCQs
- 110. Organization and Knowledge Management MCQs
- 111. Telecommunications and Networks in Knowledge Management MCQs
- 112. Components of a Knowledge Strategy MCQs
- 113. Advanced topics and case studies in knowledge management MCQs
- 114. Conventional Software Management MCQs
- 115. Software Management Process MCQs
- 116. Software Management Disciplines MCQs
- 117. Rural Management MCQs
- 118. Human Resource Management for rural India MCQs
- 119. Management of Rural Financing MCQs
- 120. Research Methodology MCQs
- 121. Research Methodology MCQs
- 122. IoT MCQs
- 123. Sensors and Actuators MCQs
- 124. IoT MCQs: Basics, Components, Protocols, and Applications
- 125. MCQs on IoT Protocols
- 126. IoT MCQs
- 127. INTRODUCTION Block Chain Technologies MCQs
- 128. Understanding Block chain with Crypto currency MCQs
- 129. Understanding Block chain for Enterprises MCQs
- 130. Enterprise application of Block chain MCQs
- 131. Block chain application development MCQs
- 132. MCQs on Service Oriented Architecture, Web Services, and Cloud Computing
- 133. Utility Computing, Elastic Computing, Ajax MCQs

- 134. Data in the cloud MCQs
- 135. Cloud Security MCQs
- 136. Issues in cloud computing MCQs
- 137. Introduction to modern processors MCQs
- 138. Data access optimizations MCQs
- 139. Parallel Computing MCQs
- 140. Efficient Open MP Programming MCQs
- 141. Distributed Memory parallel programming with MPI MCQs
- 142. Review of Object Oriented Concepts and Principles MCQs.
- 143. Introduction to RUP MCQs.
- 144. UML and OO Analysis MCQs
- 145. Object Oriented Design MCQs
- 146. Object Oriented Testing MCQs
- 147. CVIP Basics MCQs
- 148. Image Representation and Description MCQs
- 149. Region Analysis MCQs
- 150. Facet Model Recognition MCQs
- 151. Knowledge Based Vision MCQs
- 152. Game Design and Semiotics MCQs
- 153. Systems and Interactivity Understanding Choices and Dynamics MCQs
- 154. Game Rules Overview Concepts and Case Studies MCQs
- 155. IoT Essentials MCQs
- 156. Sensor and Actuator MCQs
- 157. IoT Networking & Technologies MCQs
- 158. MQTT, CoAP, XMPP, AMQP MCQs
- 159. IoT MCQs: Platforms, Security, and Case Studies
- 160. MCQs on Innovation and Entrepreneurship

- 161. Innovation Management MCQs
- 162. Stage Gate Method & Open Innovation MCQs
- 163. Innovation in Business: MCQs
- 164. Automata Theory MCQs
- 165. Finite Automata MCQs
- 166. Grammars MCQs
- 167. Push down Automata MCQs
- 168. Turing Machine MCQs
- 169. Database Management System (DBMS) MCQs
- 170. Relational Data models MCQs
- 171. Data Base Design MCQs
- 172. Transaction Processing Concepts MCQs
- 173. Control Techniques MCQs
- 174. DBMS Concepts & SQL Essentials MCQs
- 175. DESCRIPTIVE STATISTICS MCQs
- 176. INTRODUCTION TO BIG DATA MCQ
- 177. BIG DATA TECHNOLOGIES MCQs
- 178. PROCESSING BIG DATA MCQs
- 179. HADOOP MAPREDUCE MCQs
- 180. BIG DATA TOOLS AND TECHNIQUES MCQs
- 181. Pattern Recognition MCQs
- 182. Classification Algorithms MCQs
- 183. Pattern Recognition and Clustering MCQs
- 184. Feature Extraction & Selection Concepts and Algorithms MCQs
- 185. Pattern Recognition MCQs
- 186. Understanding Cybercrime Types and Challenges MCQs
- 187. Cybercrime MCQs

- 188. Cyber Crime and Criminal justice MCQs
- 189. Electronic Evidence MCQs
- 190. Ethical Hacking MCQs
- 191. Introduction to Information Security MCQ
- 192. Computer Graphics Multimedia PYQ
- 193. Style sheets MCQs
- 194. Process Control MCQS
- 195. Signals and Systems MCQs
- 196. Understanding AM and FM Transmission Noise and Receiver Characteristics
- 197. Op-Amp Characteristics MCQs
- 198. Digital filters Design Techniques Mcqs
- 199. ERROR CONTROL AND DATA LINK PROTOCOLS mcqs
- 200. Satellite Communication MCQs