

1. What is the purpose of a type system in programming languages?

- a) To enforce syntactic correctness
- b) To specify program behavior
- c) To ensure memory allocation
- d) To facilitate error detection and recovery

Answer: b) To specify program behavior

Explanation: A type system specifies the allowed operations and behavior of variables and expressions in a programming language, ensuring consistency and correctness at compile time.

2. Which component of a type checker verifies whether expressions conform to the defined types?

- a) Lexical analyzer
- b) Parser
- c) Semantic analyzer
- d) Code generator

Answer: c) Semantic analyzer

Explanation: The semantic analyzer checks whether expressions and statements conform to the types defined by the language, ensuring type correctness.

3. What does equivalence of expressions in a type system refer to?

- a) Similarity in syntax
- b) Similarity in behavior
- c) Similarity in memory allocation
- d) Similarity in compilation time

Answer: b) Similarity in behavior

Explanation: Equivalence of expressions refers to expressions that have the same behavior or produce the same results, regardless of their specific syntactic forms.

4. Type conversion in programming languages involves:

- a) Changing the syntax of a variable
- b) Altering the behavior of an expression
- c) Converting a variable from one type to another
- d) Adjusting the memory allocation of a variable

Answer: c) Converting a variable from one type to another

Explanation: Type conversion involves changing the data type of a variable or expression to facilitate operations between different types.

5. What is overloading of functions and operations in programming languages?

- a) Assigning multiple types to a single variable
- b) Defining multiple functions with the same name but different parameter types
- c) Limiting the number of operations that can be performed on a variable
- d) Restricting the assignment of values to variables

Answer: b) Defining multiple functions with the same name but different parameter types

Explanation: Overloading allows multiple functions or operations to share the same name but differ in parameter types or arity.

6. Polymorphic functions in programming languages:

- a) Are restricted to a single type
- b) Can operate on values of different types
- c) Can only be called once
- d) Are limited to a specific number of parameters

Answer: b) Can operate on values of different types

Explanation: Polymorphic functions can accept arguments of different types, allowing for greater flexibility and code reuse.

7. What aspect of storage organization deals with how variables and data are allocated and managed during program execution?

- a) Type checking
- b) Storage allocation strategies
- c) Error detection
- d) Symbol table

Answer: b) Storage allocation strategies

Explanation: Storage allocation strategies determine how memory is allocated and managed

for variables and data structures during program execution.

8. Which storage allocation strategy assigns memory to variables at compile time?

- a) Dynamic allocation
- b) Static allocation
- c) Stack allocation
- d) Heap allocation

Answer: b) Static allocation

Explanation: Static allocation assigns memory to variables at compile time, and their memory locations remain fixed throughout program execution.

9. Parameter passing refers to:

- a) Allocating memory for function parameters
- b) Passing arguments to a function
- c) Passing return values from a function
- d) Storing function parameters in a symbol table

Answer: b) Passing arguments to a function

Explanation: Parameter passing involves providing arguments to a function when calling it, which are then used as parameters within the function.

10. Dynamic storage allocation involves:

- a) Allocating memory for variables at compile time
- b) Allocating memory for variables at runtime
- c) Allocating memory for functions
- d) Allocating memory for constants

Answer: b) Allocating memory for variables at runtime

Explanation: Dynamic storage allocation involves allocating memory for variables and data structures during program execution, typically using functions like `malloc()` or `new` in languages like C and C++.

11. What purpose does a symbol table serve in a compiler or interpreter?

- a) Storing function definitions
- b) Managing memory allocation
- c) Storing information about identifiers in the program
- d) Handling runtime errors

Answer: c) Storing information about identifiers in the program

Explanation: A symbol table maintains information about identifiers (variables, functions, etc.) in the program, including their names, types, and scopes.

12. Error detection and recovery in compilers involves:

- a) Detecting and correcting syntax errors
- b) Detecting and correcting semantic errors
- c) Detecting and recovering from runtime errors

d) Detecting and reporting errors during compilation

Answer: d) Detecting and reporting errors during compilation

Explanation: Error detection and recovery in compilers involve identifying errors in the source code during compilation and reporting them to the user for correction.

13. Ad-Hoc methods of error recovery in compilers refer to:

- a) Using predefined algorithms to handle errors
- b) Handling errors on a case-by-case basis
- c) Ignoring errors during compilation
- d) Automatically correcting errors

Answer: b) Handling errors on a case-by-case basis

Explanation: Ad-Hoc methods involve handling errors based on specific conditions or situations encountered during compilation, rather than relying on predefined algorithms.

14. Systematic methods of error recovery in compilers involve:

- a) Ignoring errors and continuing compilation
- b) Using predefined algorithms to handle errors
- c) Correcting errors automatically
- d) Reporting errors without attempting recovery

Answer: b) Using predefined algorithms to handle errors

Explanation: Systematic methods of error recovery involve following predefined algorithms or strategies to handle errors encountered during compilation.

15. Which of the following statements about type systems is true?

- a) Type systems only enforce syntactic correctness
- b) Type systems specify program behavior
- c) Type systems don't facilitate error detection
- d) Type systems only ensure memory allocation

Answer: b) Type systems specify program behavior

Explanation: Type systems specify the behavior and constraints of variables and expressions in a program, ensuring consistency and correctness.

16. In parameter passing, which method involves passing the actual value of the argument to the function?

- a) Pass by reference
- b) Pass by value
- c) Pass by pointer
- d) Pass by name

Answer: b) Pass by value

Explanation: Pass by value involves passing a copy of the argument's value to the function, leaving the original value unchanged.

17. Which of the following is NOT a storage allocation strategy?

- a) Static allocation
- b) Dynamic allocation
- c) Stack allocation
- d) Register allocation

Answer: d) Register allocation

Explanation: Register allocation is a technique used by compilers to optimize the use of processor registers, rather than a storage allocation strategy.

18. What does “equivalence of expressions” mean in the context of type systems?

- a) Expressions that look the same
- b) Expressions that have the same type
- c) Expressions that produce the same result
- d) Expressions that have the same memory address

Answer: c) Expressions that produce the same result

Explanation: Equivalence of expressions refers to expressions that produce the same result or behavior, regardless of their specific syntax or form.

19. Which of the following storage allocation strategies assigns memory to variables on a last-in, first-out basis?

- a) Static allocation



- b) Dynamic allocation
- c) Stack allocation
- d) Heap allocation

Answer: c) Stack allocation

Explanation: Stack allocation allocates memory for variables in a last-in, first-out manner, typically using a stack data structure.

20. What role does the semantic analyzer play in a compiler?

- a) It checks for syntax errors
- b) It optimizes the generated code
- c) It verifies type correctness
- d) It generates machine code

Answer: c) It verifies type correctness

Explanation: The semantic analyzer checks whether expressions and statements in the program conform to the types defined by the language, ensuring type correctness.

Related posts:

1. Introduction to compiling & Lexical Analysis MCQs
2. Syntax Analysis & Syntax Directed Translation 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. Internet of Things MCQS
- 191. Analysis Design of Algorithm MCQ
- 192. Symmetric Key Cryptography MCQ
- 193. XML MCQs
- 194. System Security MCQs.



- 195. Linear Time- Invariant Systems mcqs
- 196. Control System MCQs: Basics, Feedback, and Analysis
- 197. OP-AMP applications MCQs
- 198. Radiation mcqs
- 199. NETWORKS mcqs
- 200. Satellite Services MCQs