- 1. What is the purpose of inheritance in object-oriented programming?
- a) To allow a class to inherit properties and behaviors from another class.
- b) To create new classes from existing ones.
- c) To promote code reusability and maintainability.
- d) All of the above.

Answer: d) All of the above.

Explanation: Inheritance enables the creation of new classes (derived or child classes) that inherit properties and behaviors from existing classes (base or parent classes), promoting code reusability and maintainability.

2. Which of the following is not a type of inheritance?

- a) Single inheritance
- b) Multiple inheritance
- c) Hybrid inheritance
- d) Circular inheritance

Answer: d) Circular inheritance

Explanation: Circular inheritance is not a valid type of inheritance. It refers to a situation where a class is derived from itself or from a class hierarchy where one class is derived from another in a circular manner, which is not allowed in most programming languages.

3.In the context of inheritance, what does the 'is a' relationship signify?

a) It signifies a relationship between classes where one class is a specialized version of another.

b) It signifies a relationship between classes where one class contains an instance of another.

c) It signifies a relationship between classes where both classes are equivalent.

d) It signifies a relationship between classes where one class is composed of another.

Answer: a) It signifies a relationship between classes where one class is a specialized version of another.

Explanation: The 'is a' relationship, also known as an inheritance relationship, indicates that one class (child class) is a specialized version of another class (parent class), inheriting its properties and behaviors.

4. What is association in object-oriented programming?

a) It represents a strong relationship between classes where one class contains an instance of another.

b) It represents a weak relationship between classes where one class is a specialized version of another.

c) It represents a relationship between classes where both classes are equivalent.

d) It represents a relationship between classes where one class is composed of another.

Answer: a) It represents a strong relationship between classes where one class contains an instance of another.

Explanation: Association signifies a relationship between classes where one class contains an instance of another class. It can be one-to-one, one-to-many, or many-to-many.

5. What is aggregation in object-oriented programming?

a) It represents a strong relationship between classes where one class contains an instance of another.

b) It represents a weak relationship between classes where one class is a specialized version of another.

c) It represents a relationship between classes where both classes are equivalent.

d) It represents a relationship between classes where one class is composed of another.

Answer: d) It represents a relationship between classes where one class is composed of another.

Explanation: Aggregation signifies a relationship between classes where one class is composed of another class. It represents a whole-part relationship, where the part can exist independently of the whole.

6.Which of the following best describes the concept of interfaces in object-oriented programming?

a) Interfaces define the implementation of methods and properties for a class.

b) Interfaces provide a blueprint for creating objects but cannot contain implementation.

c) Interfaces are used to inherit properties and behaviors from another class.

d) Interfaces represent a type of inheritance relationship between classes.

Answer: b) Interfaces provide a blueprint for creating objects but cannot contain implementation.

Explanation: Interfaces define a contract for classes to follow, specifying methods and properties that must be implemented by any class that implements the interface. However, interfaces themselves do not contain any implementation.

7.In object-oriented programming, what is an abstract class?

a) A class that cannot be instantiated and may contain abstract methods.

b) A class that can be instantiated but may contain abstract methods.

c) A class that cannot be inherited but may contain abstract methods.

d) A class that can be inherited but cannot contain abstract methods.

Answer: a) A class that cannot be instantiated and may contain abstract methods. Explanation: An abstract class is a class that cannot be instantiated on its own and may contain abstract methods, which are declared but not implemented in the abstract class. Abstract classes are designed to be subclassed.

8. Which of the following is not a type of inheritance?

- a) Multilevel inheritance
- b) Hierarchical inheritance
- c) Sequential inheritance
- d) Hybrid inheritance

Answer: c) Sequential inheritance

Explanation: Sequential inheritance is not a recognized type of inheritance. The common types include single, multiple, multilevel, hierarchical, and hybrid inheritance.

9. What does the 'has a' relationship represent in object-oriented programming?

a) It signifies a relationship between classes where one class is a specialized version of another.

b) It signifies a relationship between classes where one class contains an instance of another.

c) It signifies a relationship between classes where both classes are equivalent.

d) It signifies a relationship between classes where one class is composed of another.

Answer: b) It signifies a relationship between classes where one class contains an instance of another.

Explanation: The 'has a' relationship represents a composition relationship between classes, where one class contains an instance of another class.

- 10.Which of the following is true about multiple inheritance?
- a) It allows a class to inherit properties and behaviors from multiple classes.
- b) It is supported by most programming languages.
- c) It may lead to the diamond problem.

d) All of the above.

Answer: d) All of the above.

Explanation: Multiple inheritance enables a class to inherit properties and behaviors from multiple classes, but it may lead to the diamond problem, where a class inherits from two classes that have a common ancestor, causing ambiguity.

11.Inheritance in object-oriented programming allows for:

- a) Reusability of code.
- b) Redundancy of code.
- c) Limited code functionality.
- d) None of the above.

Answer: a) Reusability of code.

Explanation: Inheritance promotes code reusability by allowing new classes to inherit properties and behaviors from existing classes.

- 12. Which of the following statements about abstract classes is true?
- a) Abstract classes cannot have concrete methods.
- b) Abstract classes can be instantiated.
- c) Abstract classes cannot have constructors.
- d) Abstract classes can only have static methods.

Answer: a) Abstract classes cannot have concrete methods.

Explanation: Abstract classes can have both abstract methods (without implementation) and concrete methods (with implementation), but they cannot be instantiated on their own.

13. Which type of inheritance involves a class being derived from multiple base classes?

- a) Single inheritance
- b) Multiple inheritance
- c) Multilevel inheritance
- d) Hierarchical inheritance

Answer: b) Multiple inheritance

Explanation: Multiple inheritance involves a class inheriting properties and behaviors from multiple base classes.

14.In association, the relationship between two classes is typically represented by:

- a) Solid line with an arrowhead.
- b) Dashed line with an arrowhead.
- c) Dotted line without arrowheads.
- d) Solid line without arrowheads.

Answer: d) Solid line without arrowheads.

Explanation: In association, the relationship between two classes is represented by a solid line without arrowheads.

15. Which type of relationship between classes signifies a "whole-part" relationship?

- a) Association
- b) Aggregation

c) Inheritance

d) Composition

Answer: d) Composition

Explanation: Composition represents a "whole-part" relationship between classes, where one class contains another class as a part.

16.Interfaces in object-oriented programming are used to:

- a) Implement methods and properties.
- b) Provide a blueprint for classes.
- c) Define constructors.
- d) Inherit from other classes.

Answer: b) Provide a blueprint for classes.

Explanation: Interfaces provide a blueprint for classes, specifying methods and properties that must be implemented by any class that implements the interface.

- 17.What is the main purpose of aggregation?
- a) Code reuse.
- b) Representing "whole-part" relationships.
- c) Achieving polymorphism.
- d) Enforcing encapsulation.

Answer: b) Representing "whole-part" relationships.

Explanation: Aggregation is mainly used to represent "whole-part" relationships between classes, where one class contains another class as a part.

18. Which of the following is not a feature of an abstract class?

- a) It can have constructors.
- b) It can be instantiated.
- c) It can have abstract methods.
- d) It can have concrete methods.

Answer: b) It can be instantiated.

Explanation: Abstract classes cannot be instantiated on their own; they are meant to be subclassed.

- 20.What problem may occur with multiple inheritance?
- a) Code duplication.
- b) Inability to reuse code.
- c) Ambiguity in method resolution.
- d) Inflexibility in class design.

Answer: c) Ambiguity in method resolution.

Explanation: Multiple inheritance may lead to ambiguity in method resolution, especially in cases where a class inherits from multiple classes that have methods with the same name and signature.

21.Which type of relationship in object-oriented programming indicates a "kind-of" relationship?

- a) Association
- b) Aggregation
- c) Inheritance
- d) Composition

Answer: c) Inheritance

Explanation: Inheritance signifies a "kind-of" relationship, where a subclass is a specialized version of its superclass, inheriting its properties and behaviors.

Related posts:

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

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

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

- 103. Multimedia MCQs
- 104. Introduction to compiling & Lexical Analysis MCQs
- 105. Syntax Analysis & Syntax Directed Translation MCQs
- 106. Type Checking & Run Time Environment MCQs
- 107. Code Generation MCQs
- 108. Code Optimization 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. Advanced Computer Architecture MCQ
- 191. Introduction to Information Security MCQ
- 192. Computer Graphics Multimedia PYQ
- 193. HTML MCQs
- 194. Basic concepts of OOP MCQS
- 195. File System MCQs
- 196. Social Issues and the Environment mcqs
- 197. FM Modulation & Transmission MCQs
- 198. Introduction to ICs and Op-Amps MCQs
- 199. Efficient Computation of the DFT mcqs
- 200. OSI model mcqs