**MCQ FOR OOPS**

1. Which of the following language was developed as the first purely object programming language?
2. SmallTalk
3. C++
4. Kotlin
5. Java
6. Who developed object-oriented programming?
7. Adele Goldberg
8. Dennis Ritchie
9. Alan Kay
10. Andrea Ferro
11. Which of the following is not an OOPS concept?
12. Encapsulation
13. Polymorphism
14. Exception
15. Abstraction
16. Which feature of OOPS described the reusability of code?
17. Abstraction
18. Encapsulation
19. Polymorphism
20. Inheritance
21. Which of the following language supports polymorphism but not the classes?
22. C++ programming language
23. Java programming language
24. Ada programming language
25. C# programming language
26. Which among the following feature is not in the general definition of OOPS?
27. Modularity
28. Efficient Code
29. Code reusability
30. Duplicate or Redundant Data
31. Which feature of OOPS derives the class from another class?
32. Inheritance
33. Data hiding
34. Encapsulation
35. Polymorphism
36. Define the programming language, which does not support all four types of inheritance?
37. Smalltalk
38. Kotlin
39. Java
40. C++
41. A single program of OOPS contains \_\_\_\_\_\_\_ classes?
42. Only 1
43. Only 999
44. Only 100
45. Any number
46. Which two features of object-oriented programming are the same?
47. Abstraction and Polymorphism features are the same
48. Inheritance and Encapsulation features are the same
49. Encapsulation and Polymorphism features are the same
50. Encapsulation and Abstraction
51. Which of the following definition is incorrect for polymorphism?
52. Polymorphism helps in redefining the same functionality
53. Polymorphism concept is the feature of object-oriented programming (OOP)
54. It always increases the overhead of function definition
55. Ease in the readability of the program
56. Which among the following cannot be used for the concept of polymorphism?
57. Static member function
58. Constructor Overloading
59. Member function overloading
60. Global member function
61. Which function best describe the concept of polymorphism in programming languages?
62. Class member function
63. Virtual function
64. Inline function
65. Undefined function
66. Which member function is assumed to call first when there is a case of using function overloading or abstract class?
67. Global function
68. Local function
69. Function with lowest priority
70. Function with the highest priority
71. Is it true to use polymorphism in the Java programming language?
72. True
73. False
74. Which of the following language uses the classes but not the polymorphism concept?
75. Procedure Oriented language
76. Object-based language
77. Class-based language
78. If classes are used, then the polymorphism concept will always be used in the programming languages.
79. Which of the following OOP concept is not true for the C++ programming language?
80. A class must have member functions
81. C++ Program can be easily written without the use of classes
82. At least one instance should be declared within the C++ program
83. C++ Program must contain at least one class
84. What is the extra feature in classes which was not in the structures?
85. Member functions
86. Data members
87. Public access specifier
88. Static Data allowed
89. How many types of polymorphism in the Java programming language?
90. Three types of polymorphism
91. Two types of polymorphism
92. Five types of polymorphism
93. Four types of polymorphism
94. Which of the following feature is also known as run-time binding or late binding?
95. Dynamic typing
96. Dynamic loading
97. Dynamic binding
98. Data hiding
99. Which among the following is not a member of the class?
100. Virtual function
101. const function
102. Static function
103. Friend function
104. Which of the following class is known as the generic class?
105. Final class
106. Template class
107. Abstract class
108. Efficient code
109. Which of the following OOP concept binds the code and data together and keeps them secure from the outside world?
110. Polymorphism
111. Inheritance
112. Abstraction
113. Encapsulation
114. Which member of the superclass is never accessible to the subclass?
115. Public member
116. Protected member
117. Private member
118. All the mentioned
119. What is the size of a class?
120. Sum of the size of all inherited variables along with the variables of the same class
121. The size of the class is the largest size of the variable of the same class
122. Classes in the programming languages do not have any size
123. Sum of the size of all the variables within a class
124. Which class cannot create its instance?
125. Parent class
126. Nested class
127. Anonymous class
128. Abstract class
129. Encapsulation adds the function in a user-defined structure.
130. True
131. False
132. Which of the following variable violates the definition of encapsulation?
133. Array variables
134. Local variables
135. Global variables
136. Public variables
137. How can the concept of encapsulation be achieved in the program?
138. By using the Access specifiers
139. By using the concept of Abstraction
140. By using only private members
141. By using the concept of Inheritance
142. The concept of encapsulation helps in writing which type of classes in the Java programming language?
143. Abstract classes
144. Wrapper classes
145. Mutable classes
146. Immutable classes
147. Encapsulation is \_\_\_\_\_?
148. technique of combining more than one member functions into a single unit
149. mechanism of combining more than one data member into a single unit
150. mechanism of combining more than one data members and member functions that implement on those data members into a single unit
151. technique of combining more than one data members and member functions into a single unit, which can manipulate any data
152. Which of the following statement of a program is not right?
153. class teacher{ }; teacher s[5];
154. class teacher{ }s;
155. class teacher{ }; teacher s;
156. class teacher{ }s[];
157. Which of the following syntax is incorrect for the class definition?
158. student class{ };
159. class student{ student(int a){} };
160. class teacher{ public: teacher(int a){ } };
161. None of the mentioned
162. In Java the object cannot be \_\_\_\_\_\_\_\_?
163. passed by copy
164. passed as function
165. passed by value
166. passed by reference
167. Which among the following feature does not come under the concept of OOPS?
168. Data binding
169. Data hiding
170. Platform independent
171. Message passing
172. Which of the following feature may be breaken if the user does not use the classes in the code?
173. Object must be used violated
174. Only the encapsulation concept is violated
175. Inheritance cannot be implemented
176. Basically, all the features of OOPS get violated
177. Which of the following feature interacts one object with another object?
178. Message reading
179. Message passing
180. Data transfer
181. Data binding
182. Which definition best defines the concept of abstraction?
183. Hides the important data
184. Hides the implementation and showing only the features
185. Hiding the implementation
186. Showing the important data
187. The combination of abstraction of the data and code is viewed in \_\_\_\_\_\_\_\_.
188. Inheritance
189. Object
190. Class
191. Interfaces
192. The principle of abstraction\_\_\_\_\_\_\_\_\_\_\_
193. is used to achieve OOPS
194. is used to avoid duplication
195. use abstraction at its minimum
196. is used to remove longer codes
197. Which among the following concept is correct if a user using the concept of encapsulation in a code?
198. The modification of the code can be additional overhead
199. Member functions can be used for modifying the data type of data members
200. The data type of data member cannot be modified
201. The data type of the data member can be easily modified without modifying any other code
202. Using the concept of encapsulation security of the data is \_\_\_\_\_\_\_\_\_\_\_.
203. Ensured to some extent
204. Purely ensured
205. Not ensured
206. Very low
207. The name of the default access specifier for the member functions or data members in the Java programming language is \_\_\_\_\_\_\_\_\_.
208. Private access specifier
209. Public access specifier
210. Protected access specifier
211. Default access specifier
212. What is the implicit return type of constructor?
213. No return type
214. A class object in which it is defined
215. void
216. None
217. When is the object created with a new keyword?
218. At run time
219. At compile time
220. Depends on the code
221. None
222. Identify the incorrect constructor type.
223. Friend constructor
224. Default constructor
225. Parameterized constructor
226. Copy constructor
227. Choose the option below which is not a member of the class.
228. Friend function
229. Static function
230. Virtual function
231. Const function
232. What is the number of parameters that a default constructor requires?
233. 0
234. 1
235. 2
236. 3
237. In Java the undefined access modifier of a method is private?
238. True
239. False
240. Depends on code
241. None
242. Which of the following functions can be inherited from the base class?
243. Constructor
244. Destructor
245. Static
246. None
247. Which of the following is not a type of inheritance?
248. Multiple
249. Multilevel
250. Distributed
251. Hierarchical
252. What is an object in Java?
253. It is a function of class
254. It is an instance of the class
255. It is a data type of the class
256. It is a part of the syntax of the class
257. Why is reusability a desirable feature?
258. Reduces compilation time
259. Decreases testing time
260. Lowers maintenance cost
261. None
262. Another name of overloading is?
263. Pseudo polymorphism
264. Transient polymorphism
265. Virtual polymorphism
266. Ad-hoc polymorphism
267. Functions or methods of parent class is redefined in the child class is known as:
268. Overloading of methods
269. Overriding of methods
270. Redefinition of methods
271. Restoration of methods
272. Identify the feature, which is used to reduce the use of nested classes.
273. Binding
274. Inheritance
275. Encapsulation
276. Abstraction
277. How do encapsulation and abstraction differ?
278. Hiding and binding
279. Binding and hiding
280. Hiding and holding
281. None
282. Choose the option below which is shown by function overriding
283. Abstraction
284. Encapsulation
285. Polymorphism
286. Inheritance
287. Choose the option below for which instance of the class cannot be created.
288. Anonymous class
289. Parent class
290. Nested class
291. Abstract class
292. Which of the following is not an oops concept?
293. Inheritance
294. Compilation
295. Polymorphism
296. Encapsulation