# Test your OOP Skills in C++

## True and False Questions

Please check whether the following statements are true or false.

1. The % operator cannot be used between two float values.
2. Other than zero, every number specifies true value for conditional statements.
3. A sizeof() operator cannot give the size of structure variable.
4. In implicit data type conversion, the values are converted to the type of the largest operand in an expression.
5. Every statement in C++ is terminated by the semicolon.
6. Conditional operators have higher precedence than the the assignment operators.
7. switch expression cannot be of float type.
8. do-while statement terminates with the semicolon.
9. We cannot use multiple conditions in for loop.
10. switch statement cannot be nested.
11. We cannot declare variables in the for loop.
12. break statement terminates the program.
13. We cannot read spaces with cin>>.
14. cin and cout are the objects of iostream class.
15. All preprocessor directives start with # sign.
16. An array can be a collection of dissimilar elements also.
17. A structure can be a collection of similar elements also.
18. We cannot change the base address.
19. Static binding means that the code associated with a given procedure is known at the time of the call at compile time.
20. Array name always refers to the base address.
21. Function prototype is always required when we use the functions in a program.
22. void functions cannot be used in the expression.
23. In procedure-oriented programming, all data are shared by all functions.
24. We can pass the parameters to the function by using call by pointers.
25. The parameters associated with the function call are called actual parameters.
26. By default C++ assumes the return type of any function as void.

27. By default C++ assumes the storage specifiers of all variables as static.
28. C++ was developed by Denis Ritchi.
29. Data members of a class cannot be initialized inside class specifiers.
30. By using #define we can declare constants.
31. Strings are always terminated by the NULL character.
32. Pointer is also a variable that can hold the address of another variable.
33. The only purpose of getch() is to hold the screen.
34. Recursion is a method in which function calls itself.
35. A class is an instance of an object.
36. Friend functions cannot be used to overload assignment operators.
37. Inheritance provides the code reusability.
38. An identifier can start with an underscore.
39. Escape sequences always starts by \.
40. A derived class is never used to create objects.
41. "a" is a string of two characters – a and NULL.
42. The combination of data and the functions is called data abstraction.
43. We can use multiple initializations in for loop.
44. Call by reference method changes the value of actual parameters if the formal parameters are changed.
45. In procedure-oriented programming approach data is not secured.
46. The procedure-oriented programming focuses on the action.
47. In procedure-oriented programming the program is divided into several functions.
48. Object-oriented programming focuses on the data rather than functions.
49. Structure variables are also called as structure tag.
50. We cannot use functions in the structures.
51. int A[4]={1,2,3,4}; Suppose the base address is 1000 then cout<<(A+1); will display 1001.
52. A[2] is equivalent to 2[A] in C++.
53. Dot operator is used to access the structure elements through pointers.
54. A[0][0][0] is equivalent to ***A in a three-dimensional array.
55. A structure can contain another structure.
56. In C++, all members of a class are public by default.
57. Default arguments are used when the trailing arguments are missing in a function call.
58. We can use comma operator in for loop.
59. By default all structures members are private.
60. Scope resolution operator is used to access the structure member through pointer to structure?
61. An array can contain structure types of elements.
62. A function cannot be declared a friend of more than one class.

63. Member functions defined outside the class are inline functions by default.

64. Friend function in the class can be declared only in the public mode.

65. We can access the friend function of a class through the object by using dot operator.

66. A class is an abstract data type.

67. When the parameters are passed by call by reference then Changes are not reflected into the actual parameters.

68. A friend of a class can access all the private and protected members of a class directly.

69. The private members of the class implement the concept of data hiding of OOP.

70. A friend function of a class can use the private and protected members of the same class through the object of the same class.

71. The member functions of a class can be declared both inside and outside the class.

72. The class declaration is always terminated by the semicolon.

73. A derived class establish an "Is-A" relationship with the base class.

74. A for loop can never be terminated by the semicolon.

75. Class is a blue print of all the objects having same state and different behavior.

76. Memory for object is allocated when the class is declared.

77. Memory for the member functions of a class is allocated when the object is created.

78. Destroy operator is used for dynamically de-allocating the memory?

79. A function can return more than one value.

80. A function can return an object.

81. A static member function can access only static members of the same class.

82. private, public and protected are called access specifiers.

83. Memory is wasted when we use inline function.

84. NULL pointer is also called a zero pointer?

85. The private data and member functions of a class can be accessed only by the member function or friend function of the same class.

86. Calling a function is also called message sending.

87. We can initialize the data members of a class at the time of their declaration.

88. Constructor should have the same name as of the class name.

89. Constructor cannot return any value.

90. We can pass the parameters to the constructor.

91. When base class is inherited in private mode by the derived class all member functions of th derived class become private to the derived classes.

92. A constructor can be virtual.

93. A constructor can be inherited.

94. A destructor called the delete operator.

95. A destructor can be virtual.

96. We can pass the parameters to the destructor also.

97. There are three types of destructors – default, parameterized and copy destructor.

98. Function overloading can be implemented with the different return types of the functions with the same signature.

99. Function name along with the parameters is also called signature of the function.

100. Constructors cannot be overloaded.

101. Destructors can be overloaded.

102. Compile time polymorphism is implemented by function overloading.

103. In hierarchical inheritance, more than one class can be derived from a single base class.

104. The for loop is best when we know the fixed number of iterations.

105. Inheritance is used to implement generalization and specialization.

106. Private members of a base can be inherited in the private mode only.

107. Public members of a base class become private members of the derived class in private mode inheritance.

108. The private, public and protected are called the visibility mode in inheritance.

109. Protected members can be inherited but private members cannot be inherited.

110. A container class contains both the friend and the inline functions.

111. A nested class contains all the member functions in public mode.

112. A nested class contains the objects of another class as data members.

113. Protected members can be inherited in private mode only.

114. Virtual base class is used to remove ambiguity in hybrid inheritance.

115. A class cannot be a member of another class.

116. An object of one class can be a member of another class.

117. A class that contains another class is called nested class.

118. A class that contains objects of some other class is called container class.

119. A class that contains objects of some other class is called abstract class.

120. A class from which another class inherits its properties is called super class.

121. When a class inherits from more than one class it is called multiple inheritance.

122. The derived class can directly access the private members of the base class.

123. The base class can access all the members of the derived class.

124. We can perform only addition and subtraction on pointer variables.

125. & and * have higher precedence than arithmetic operators.

126. if int *ptr, then ptr++ will point to the next integer number.

127. The delete operator releases memory dynamically allocated through the new operator.

128. We can declare an array of pointers.

129. float a=10/3; will give 3.333.

130. An array is always a collection of integer data item.

131. Memory is allocated during function definition.

132. We can pass the reference to the constant value also.

133. stdio.h header file is required to use gets().

134. A function call can be used in an expression also.

135. A structure can contain the pointer which points to structure itself.

136. When we call a member function of a class, this pointer is automatically passed.

137. Array elements are accessed by using dot operator.

138. To access the public members of a class with a pointer to an object, an arrow operator is used.

139. A stream may be connected to more than one file at a time.

140. The << is called an insertion operator.

141. struct is a user-defined variable.

142. A=B is valid if A and B are the variables of the same structure type.

143. While accessing the structure members, left side of dot operator must be structure pointer.

144. Inline functions are used for a large set of code.

145. The function declaration is also called function prototype.

146. The constructor which does not take any parameter is called a default argument constructor.

147. We can define the function before the function call.

148. Function prototype in not required when we define the function before its call.

149. A class can have public data and some private functions.

150. A virtual function can be defined as a friend function of another class.

151. Static storage specifiers retain the value between the function calls.

152. The scope resolution operator cannot be overloaded.

153. A pure virtual function is a type of function which has only a function declaration.

154. The ternary operator (? :) can be overloaded.

155. A class which contains the pure virtual function is called abstract class.

156. The inheritance can be applied up to level 6 only.

157. <stdio.h> header file is required to use srtcat().

158. & is an address of operator.

159. Lifetime of extern variable is a file.

160. Parameters associated with function definition are called formal parameters.

161. Pointer can contain the address of another pointer.

162. The extraction operator is a member of istream class.

163. If the member function has not been declared as virtual, the base class member function is always called through the pointer to base class because linking takes place during compile time.

164. A dynamic binding can be achieved by virtual function.

165. A pointer to base class can hold the address of derived class only.

166. Operator overloading can be done by using friend function.

167. The unique name of the class is called its signature.

168. In a copy constructor, the argument can be passed by reference.

169. A negative number is evaluated to true in the logical expression in C++.

170. An abstract base class contains at least one pure virtual function.

171. A class can have virtual constructors.

172. We cannot create an object of the virtual base class.

173. We cannot create an object of the abstract class.

174. The private members of the base class cannot be inherited.

175. void is a data type.

176. A class can have many constructors.

177. This pointer points to the object that is currently used to invoke a member function of a class.

178. A base class is a generalized class.

179. The C + + programming language was invented by Bjarne Stroustrup.

180. Encapsulation is a bundling of data and functions together.

181. The ability of a function or operator to act in different ways on different data types is called overloading.

182. A class can have many destructors.

183. The inline functions save the execution time but take more memory space.

184. In call by reference, address of an object is passed as an argument to a function.

185. Class consists of data and member functions and structure consists data only.

186. The destructors are executed in reverse order (from most derived class to base class) in inheritance.

187. Compile-time polymorphism is also called as late binding.

188. When no constructor is defined in the class then compiler supplies the default constructor.

189. A constructor is called using new operator.

190. Scope resolution operator is used to define the outline member function.

191. Forward declaration is used when a member function of one class is a friend of another class.

192. A member function of one class can be a friend of another class.

193. The goto statement transfers the control to a label.

194. We can override a function template for particular type.

195. We cannot inherit a new class from the class template.

196. A function template can have multiple argument types.

197. Template class can be defined for user-defined data types.

198. Template argument can take default values.

199. virtual keyword in the virtual function definition can appear before return type.

200. Operator overloading is a concept by which we can design new object.

201. A pure virtual function is a virtual function that has no body.

202. An int data type requires 4 bytes of memory storage.

203. The statement if (1/2) {cout<< "hello" ;} else {cout<< "hi" ;} will display hello.

204. Private member function can access the private members of the same class?

205. Private access mode implements data abstraction concept of OOP.

206. All member functions defined inside the class are virtual functions.

207. The inheritance is used to change the visibility modes.

208. A virtual function cannot have a constructor member function but it can have destructor member function.

209. All member functions defined inside the class are inline functions.

210. Friend function can be defined in any access mode of a class.

211. this pointer holds the address of a class.

212. A friend function of a class cannot access the public members of a class?

213. The virtual function cannot alter the data members.

214. const function cannot alter the data members.

215. The class without a tag name is called anonymous class.

216. The class without a tag name is called container class.

217. get() and put() are used to read and write block of data.

218. Constructor is a friend function of a class.

219. Constructor can be defined in private mode also.

220. Destructor is called automatically after the constructor.

221. Encapsulation, Inheritance and Polymorphism are the main features of OOP.

222. Constructor can also return any data of user-defined type.

223. Inheritance is a relationship in which a class includes one or more objects of another class.

224. The inheritance is used to avoid rewriting of the code.

225. The ambiguity in the single-level inheritance is removed by using colon operator.

226. In C++, we can inherit data members, member functions and friend function only.

227. Inheritance is very useful because it provides extension.

228. A base class is a specialized class.

229. A file in C++ can be opened by using open() function.

230. << operator is used to send output to an output file.

231. State properly what distinguishes one object from the others.

232. A pure virtual function is a virtual function that defines an abstract class.

233. In operator overloading we can change the template of the operator.

234. Only one parameter is required, when unary operator is overloaded using friend function.

235. Object cannot be created of a class containing pure virtual function.

236. void operator ++ (int) is used to overload pre-increment ++.

237. Only one parameter is required to overload the binary operator through a friend function.

**238.** Template function keyword is used to define a function template in C++.

**239.** Template is used for nested class.

**240.** Object-oriented programming language supports inheritance.

**241.** Template is used for container class.

**242.** Template class can be inherited.

**243.** Object-based languages support only class, object and inheritance.

**244.** Polymorphism supports the capability of one class to use properties of another class.

**245.** Encapsulation is used to implement data abstraction.

**246.** Object-based programming language supports data abstraction.

**247.** Inheritance supports the capability of sending same message to objects of several different classes.

**248.** The polymorphism is a way for an entity to behave in several forms.

**249.** Object-based programming language supports operator overloading.

**250.** Object-based programming language makes the software reuse possible.

**251.** Exception handling in general, is a way of dealing with exceptional errors.

**252.** Generate, handled, catch are the three keywords used with exception handling.

**253.** Inheritance, templates and exception handling are the main features of OOP.

**254.** catch block is called an exception handler.

**255.** C++ programming language supports top-down and bottom-up design concept.

**256.** UML stands for unified modeling language.

**257.** The child class is called the ancestor class.

**258.** A class should have state and behavior.

**259.** fobj.seekg(0) will place the pointer to the beginning of fobj.

**260.** Object is an instance of a class.

**261.** An object should have an identity, state and behavior.

**262.** In a binary file, no character translation takes place.

**263.** A file in C++ can be opened by using constructor of the appropriate class.

**264.** fstream class is derived from iostream.

**265.** Encapsulation is the division of a program into independent modules.

**266.** Abstraction is a collection of necessary data items and function.

**267.** A base class can inherit the properties of a derived class.

**268.** Data abstraction hides the information.

**269.** Derived class is a user-defined data type.

**270.** A class is a metadata.

**271.** Rectangle box is used to draw the object diagram.

**272.** Object represents data and its associated function under single unit.

**273.** A class is a group of similar objects that do not share common properties and behavior.

274. The object name is also called as its state.

275. Selector method and modifier methods are the two types of methods in OOP.

276. An object has state, message, behavior.

277. Encapsulation hides the details of state, behavior and identity of an object.

## Answers

| | | | | |
|---|---|---|---|---|
| 1. F | 2. T | 3. F | 4. T | 5. F |
| 6. T | 7. T | 8. T | 9. T | 10. F |
| 11. F | 12. F | 13. T | 14. T | 15. T |
| 16. F | 17. T | 18. T | 19. T | 20. T |
| 21. F | 22. T | 23. T | 24. T | 25. T |
| 26. F | 27. F | 28. F | 29. T | 30. T |
| 31. T | 32. T | 33. F | 34. T | 35. F |
| 36. T | 37. T | 38. T | 39. T | 40. F |
| 41. F | 42. F | 43. T | 44. T | 45. T |
| 46. T | 47. T | 48. T | 49. F | 50. F |
| 51. F | 52. T | 53. F | 54. T | 55. T |
| 56. F | 57. T | 58. T | 59. F | 60. F |
| 61. T | 62. F | 63. F | 64. F | 65. F |
| 66. T | 67. F | 68. F | 69. T | 70. T |
| 71. T | 72. T | 73. T | 74. F | 75. F |
| 76. F | 77. F | 78. F | 79. F | 80. T |
| 81. T | 82. T | 83. T | 84. T | 85. T |
| 86. T | 87. F | 88. T | 89. T | 90. T |
| 91. F | 92. F | 93. F | 94. T | 95. T |
| 96. F | 97. F | 98. F | 99. T | 100. F |
| 101. F | 102. T | 103. T | 104. T | 105. T |
| 106. F | 107. T | 108. T | 109. T | 110. F |
| 111. F | 112. F | 113. F | 114. T | 115. F |
| 116. T | 117. T | 118. T | 119. F | 120. T |
| 121. T | 122. F | 123. F | 124. T | 125. T |
| 126. T | 127. T | 128. T | 129. F | 130. F |
| 131. T | 132. F | 133. T | 134. T | 135. T |
| 136. T | 137. F | 138. T | 139. T | 140. T |
| 141. F | 142. T | 143. F | 144. F | 145. T |

| | | | | |
|---|---|---|---|---|
| 146. F | 147. T | 148. T | 149. T | 150. F |
| 151. T | 152. T | 153. T | 154. F | 155. T |
| 156. F | 157. F | 158. T | 159. T | 160. T |
| 161. T | 162. T | 163. T | 164. T | 165. F |
| 166. T | 167. F | 168. T | 169. T | 170. T |
| 171. T | 172. F | 173. T | 174. T | 175. T |
| 176. T | 177. T | 178. T | 179. T | 180. T |
| 181. T | 182. F | 183. T | 184. T | 185. T |
| 186. T | 187. F | 188. T | 189. F | 190. T |
| 191. F | 192. T | 193. T | 194. T | 195. F |
| 196. T | 197. T | 198. T | 199. T | 200. F |
| 201. T | 202. F | 203. F | 204. T | 205. F |
| 206. F | 207. F | 208. T | 209. T | 210. T |
| 211. F | 212. F | 213. F | 214. T | 215. T |
| 216. F | 217. F | 218. F | 219. T | 220. F |
| 221. T | 222. F | 223. F | 224. T | 225. F |
| 226. F | 227. T | 228. F | 229. T | 230. T |
| 231. T | 232. T | 233. F | 234. T | 235. T |
| 236. F | 237. F | 238. T | 239. F | 240. T |
| 241. T | 242. T | 243. F | 244. F | 245. T |
| 246. T | 247. F | 248. T | 249. F | 250. F |
| 251. T | 252. F | 253. F | 254. T | 255. T |
| 256. T | 257. F | 258. F | 259. T | 260. T |
| 261. T | 262. T | 263. T | 264. T | 265. F |
| 266. T | 267. F | 268. F | 269. F | 270. T |
| 271. T | 272. F | 273. F | 274. F | 275. T |
| 276. F | 277. F | | | |