

Object Oriented Programming Through C++

Very Short Questions:

1. What is the full form of OOP?
2. Define a class.
3. What is inheritance?
4. What do you mean by message passing?
5. Define data abstraction.
6. Who developed C++?
7. What is a token?
8. What is a variable?
9. What are cin and cout?
10. What is a structure?
11. What is the difference between structure, union, and classes?
12. What is an access specifier?
13. What is the syntax for writing a destructor?
14. What is a static function?
15. What is the syntax for passing objects to functions?
16. List ways of passing arguments to a function.
17. What is runtime polymorphism?
18. What are the types of compile-time polymorphism?
19. What are different types of inheritance?
20. How is multiple inheritance resolved in C++?
21. Define abstract classes.
22. What are the types of operator overloading?
23. Define hybrid inheritance.
24. How is function overloading implemented?
25. Define a pure virtual function.
26. What do you mean by exception handling?
27. What should be placed inside a try block?
28. What should be placed inside a catch block?
29. What is an exception?
30. What are templates?
31. What is generic programming?
32. How can a normal function be declared as a template function?
33. How can a member function be declared as a template function?
34. Explain the need for templates.
35. Explain a class template.
36. Explain the working of a template function.
37. Distinguish between overloaded functions and function templates.
38. Write a program to create a template to find the maximum value stored in an array.
39. What is the difference between ios::ate and ios::app modes?
40. What are the two file pointers?

41. What is a file?
42. What is the significance of the tellg() function?
43. What are input and output streams?
44. What is sequential file access?
45. What is random access file?
46. What is looping?
47. Write syntax for a for loop.
48. Write syntax for an if-else ladder.
49. What is a break statement?
50. What is an inline function?
51. What is the difference between function overloading and function overriding?
52. List the characteristics of a friend function.
53. What is a pointer in C++?
54. What is a null pointer?
55. What is a class?
56. What is an inline function?
57. Why is a const member function used?
58. Name various types of member functions in C++.
59. What is an object?
60. What is the difference between a class and a structure?
61. In how many ways can a class be defined in C++?
62. What are member data and member functions in C++?
63. What is a scope resolution operator, and what is its usage?
64. What is a constructor?
65. What is a destructor?
66. What is the order of destructor execution in C++?
67. What is the "this" pointer?
68. What are the types of constructors?
69. When is a constructor called?
70. What is the use of the "this" pointer?
71. What is the syntax of a constructor?
72. What is an example of a destructor?
73. Why are constructors needed?
74. Which constructor is called in a derived class?
75. What is inheritance?
76. Why is a constructor required in a derived class?
77. Define multiple inheritance.
78. What is the use of constructors and destructors in a derived class?
79. What are derived classes and base classes?
80. How do you implement single inheritance in C++?
81. What is the destructor in a C++ class example?
82. What are the types of inheritance?
83. What is overloading?
84. What does polymorphism mean in C++?

85. What is static binding?
86. What is dynamic binding?
87. What is an abstract class?
88. What is function overloading?
89. What is operator overloading?
90. Why is it necessary to overload an operator?
91. What is a virtual function?
92. Why do we need virtual functions?
93. What is a template in C++?
94. Can a function template take multiple arguments in C++?
95. When should you use templates in C++?

Short Questions:

1. What are the unique advantages of OOP?
2. List the applications of OOP technology.
3. What is the scope of a class object?
4. What are the features of OOP?
5. What are the rules for naming a class?
6. Write a C++ program to find the factorial of a number.
7. Write a C++ program to find the largest digit in a number.
8. What is the difference between an array and a structure?
9. What are inline functions?
10. Explain type conversion in C++.
11. Explain objects with an example.
12. What is a destructor, explaining with an example?
13. Explain types of constructors.
14. What is the difference between a macro and an inline function?
15. Explain all types of member functions.
16. Write a program to explain the copy constructor.
17. Explain protected access specifier with an example.
18. What is a pointer to classes?
19. Define an array of objects in detail.
20. Explain inheritance with an example.
21. What is a virtual function?
22. Explain polymorphism in detail with its types.
23. Define constructors in derived classes.
24. Define destructors in derived classes.
25. Write a program to calculate the area of a circle and rectangle using function overloading.
26. Explain binary operator overloading with an example using the operator `&`.
27. Define hybrid inheritance.
28. Why do we need inheritance, and what are the disadvantages of using it?

29. What are default arguments, and how are they related to function overloading?
30. Explain the guidelines for exception handling.
31. Explain the mechanism of exception handling.
32. Describe the role of keywords `try`, `catch`, and `throw` in exception handling.
33. How many different types of errors are encountered in a program?
34. Explain specifying exceptions with necessary steps.
35. How can you manage runtime errors? Explain.
36. Differentiate between binary and text files.
37. What is the difference between opening a file with a constructor function and opening a file with the `open()` function?
38. What is the file access mode? Describe the various file modes.
39. Compare and contrast `read()`, `getline()`, and `readlines()`.
40. How is `write()` different from `writelines()`?
41. In how many ways can the end of a file be detected?
42. How many file modes can be used with the `open()` function to open a file? State the function of each mode.
43. What do the `seekg()` and `seekp()` functions do?
44. Explain the meaning of the C++ programming language.
45. Describe the following characteristics of OOP: Encapsulation, Polymorphism, Abstraction, Inheritance.
46. Compare C and C++ with examples.
47. Compare object-oriented programming with procedural programming.
48. Discuss the issues of procedure-oriented systems with respect to object-oriented systems.
49. Draw a diagram to represent the basic structure of a C++ program.
50. What is the need for data types in C++?
51. Differentiate between keywords and identifiers.
52. Explain all the data types in C++.
53. How many types of operators are there in C++? Explain.
54. Write a C++ program to display your personal information.
55. What are keywords?
56. What are access specifiers? Explain each.
57. What is the need for arrays and discuss different types of arrays.
58. Explain the syntax of a for loop in C++ with a suitable example.
59. What is the difference between an array and a list?
60. What is a static member function?
61. What do you know about exit control and entry control loops?
62. Write a short note on access specifiers in C++.
63. Define a pointer with a suitable example.
64. What is a friend function? Write the syntax of a friend function.
65. Write a program to print the address of a variable whose value is input from the user.
66. What is the usage of the pointer in C?
67. Discuss the need for a friend function using a suitable example.
68. Write a short note on virtual functions.

69. What is a class and an object in C++?
70. What is an object?
71. What is a friend function in C++?
72. What is operator overloading in C++?
73. How many ways can a class be defined in C++?
74. What is a constructor? Explain with examples.
75. How are constructors different from normal member functions?
76. What is the `this` pointer (`->`)? Explain this pointer with a program.
77. What is the difference between a constructor and a destructor?
78. Write a program for constructor overloading.
79. How is a destructor different from a member function?
80. List the types of inheritance and write a C++ program to implement single inheritance.
81. What is the importance of inheritance?
82. What is a constructor and destructor in derived classes in C++?
83. What is multiple inheritance, and write a program of multiple inheritance?
84. How is polymorphism achieved at compile time and runtime?
85. When do we make a virtual function "pure," and what are the implications of making a function a pure virtual function?
86. Differentiate between early binding and late binding.
87. Explain the concept of polymorphism.
88. What is file handling in C++?
89. What is an input file stream in C++?
90. What is an output file stream in C++?
91. How do you open a file for reading in C++?
92. How do you open a file for writing in C++?
93. What is the difference between reading a file and writing a file in C++?
94. How do you close a file in C++?
95. Differentiate between binary and text files.
96. How is `write()` different from `writelines()`?
97. In how many ways can the end of a file be detected?
98. What do the `seekg()` and `seekp()` functions do?
99. What is generic programming, and how is it implemented in C++?
100. Explain with the help of an example why templates are used in programming.
101. Can a function template take multiple arguments in C++?
102. Explain when you should use templates in C++.
103. Explain the advantages of OOP in the field of software development.
104. What is the difference between object-oriented programming and procedural programming?
105. Explain bitwise operators and increment/decrement operators.
106. What is the difference between unary and binary operators in C++?
107. What is the difference between the assignment operator and the equality operator in C++?
108. What is the difference between implicit and explicit type conversion in C++?
109. What is a 1D and 2D array in C++?

110. What is the syntax of the if-else statement in C++?
111. How do you use nested if statements in C++?
112. What is the syntax and working of the for loop in C++?
113. Differentiate between the for loop and the while loop in C++.
114. What is the do-while loop in C++?
115. Differentiate between

the do-while and while loops in C++.

116. What is the purpose of the break statement in a loop or switch case in C++?
117. Differentiate between the if-else and switch statements in C++.
118. How do you use the switch case statement to handle multiple cases in C++?
119. What is a function in C++ and explain the default argument concept.
120. How do you declare a function in C++, and explain the inline function concept?
121. Explain the various parts of a function in C++ and explain the function overloading concept.
122. Differentiate between call by value and call by reference.
123. What is aliasing of a variable, and explain the usage of the scope resolution operator `::` in C++.
124. How do you access members of a structure using the `.` and `->` operators?
125. What is the difference between a structure and a union in C++? Give an example of when you might use each.
126. What is the syntax for declaring a structure in C++? How do you access its members?
127. What is the syntax for declaring a union in C++? How do you access it?
128. How do you initialize an array in C++?
129. How do you access elements of an array in C++?
130. What is the difference between a class and an object in C++? Explain with an example.
131. How do you define member functions for a class in C++? Explain with an example.
132. How to define member functions inside and outside of the class.
133. How do you access the members of a class in C++? Explain with an example.
134. What is the difference between a default constructor and a parameterized constructor in C++?
135. What is the difference between a constructor and a destructor in C++?
136. Explain various access specifiers in C++.
137. What is the purpose of a friend function in C++?
138. Can a friend function access private data member of a class in C++?
139. What is the purpose of a friend class in C++?
140. How do you declare a friend class in C++?
141. Can a friend class access private data member of a class in C++?
142. What is the difference between a friend function and a member function in C++?
143. When should you use friend functions and friend classes in C++?
144. Explain stream function overloading using a friend function.
145. How is a static member different from a non-static member of a class in C++?
146. Can a static member function access non-static data member of a class in C++?
147. When should you use a static member of a class in C++?

148. What is the difference between a base class and a derived class in C++?
149. What is a virtual function in C++ and a pure virtual function in C++?
150. What is the purpose of a virtual function in C++?
151. What is the order of constructor and destructor calling in inheritance in C++?
152. What type of ambiguity occurs in multiple inheritance, and how to resolve it?
153. What is polymorphism in C++? Explain its two types of polymorphism in C++.
154. What is function overloading in C++ and how to achieve it in C++?
155. What is operator overloading in C++ and how is operator overloading achieved in C++?
156. What is function overriding in C++ and how to achieve it in C++?
157. What is the difference between function overloading and function overriding in C++?
158. How is a template different from a function in C++?
159. What is a function template, and how is a function template declared in C++?
160. What is a class template in C++, and how is a class template declared in C++?
161. How to handle all types of exceptions in a single catch statement?
162. Explain the advantages of exception handling in C++.
163. Explain various modes for opening a file.

Long Questions:

1. Difference between Functional Language and OOP Language.
2. Explain the evolution of OOP Language.
3. Write characteristics of OOP in detail.
4. What do you mean by dynamic binding? How it is useful in OOP.
5. Write short note on Polymorphism and overloading.
6. Explain Operators in C++
7. Explain conditional Statements with example.
8. What is multidimensional array write a C++ program for linear and binary search.
9. What is Structure? Explain with example.
10. Explain function Calling mechanisms in C++.
11. Write a program to find mean and median using classes?
12. Explain friend function in detail. Give example?
13. Write a program to illustrate Bank withdrawal and deposit process. User should be able to know his account balance in the end after transactions being made in his or her account?
14. Write a program to display the result of students of a class using an array of objects?
15. Explain this pointer with an example?
16. Write a program to explain what is multilevel inheritance?
17. Write a program to compare 2 strings using operator overloading?
18. Explain all types of operator overloading? Give an example of any 2.
19. Write a program to display details of a person issuing a book from the library using inheritance. Person information includes person_id, name, book issued, date of issuing, date of returning, book_id, address, and contact no?
20. Explain types of inheritance with an example of any one?
21. What are the differences between synchronous and asynchronous exceptions?

22. Explain the exception handling model of C++ with various constructs supported by it.
23. What are the advantages of using the exception handling mechanism in a program?
24. Explain the use of output functions write() and writeline() with an example each.
25. Write a function that writes a structure to disk and then reads it back and displays it on the screen.
26. Explain the following functions with examples:
 - i. Fstream
 - ii. Ofstream
 - iii. Ifstream
27. Describe the various classes available for file operations.
28. Consider the following statements:
Fstream file;
File.open("ABC", ios::in | ios::out);

Write C++ statement(s) for the following:

- i. To move the pointer at the beginning of the file.
 - ii. To move the pointer at the end of the file.
 - iii. To find the total number of bytes
 - iv. To close the file.
29. Explain the functioning of the following:
 - i. Fstream file;
 - ii. File.seekg (100, ios::cur);
 - iii. File.seekg (-100, ios::end);
 - iv. File.seekg (100, ios::beg);
 30. The record consists of two fields: name and rollno. Write a program that will perform the following:
 - i. Create a data file of 5 records.
 - ii. Display a data file.
 - iii. Append a record.
 - iv. Modify one of the records.
 31. Explain generations of programming languages.
 32. Explain basic concepts/properties/principle of OOPS in detail.
 33. Difference between Procedural Oriented Programming Languages (Pops) and OOPS.
 34. Write benefits of OOPS.
 35. Write applications of OOPS.
 36. Explain various categories of programming approaches.
 37. Explain the following concepts of object-oriented programming concept in detail with examples:
 - i. Data Abstraction
 - ii. Inheritance
 - iii. Polymorphism
 - iv. Object
 38. State the important features of OOPS.
 39. What are the basic concepts of OOPS language? Explain each.

40. Describe class and object concept in detail.
41. Define OOPS. Describe various characteristics of OOPS.
42. Give classification of operators available in C++ with the help of a neat and clean diagram.
43. What is the need for type conversion? Discuss different types of type conversion in C++.
44. What is the need for data types in C++? Describe different data types along with their representations and size in C++.
45. What is typecasting? What are explicit and implicit type conversions?
46. What is a user-Defined data type in C++? Explain the various user-Defined data types in C++.
47. What are Operators? Explain their various types.
48. Write a program using an array to input a minimum of 3 records (3 persons data) and display all the records.
49. What is the need for an array? Write some examples.
50. Write a program to demonstrate the concept of an array of objects.
51. Explain the various types of jumping statements in C++ with examples.
52. Write a C++ program to read n values into an array, find the sum and average using a pointer with a function.
53. Explain all the Looping statements available in C++.
54. Write a Short Note on:
 - i. C++ tokens
 - ii. Jumping statements
 - iii. Arrays
55. What is the need for an array? Discuss different types of arrays.
56. Describe an array in C++ in detail.
57. How to Define member functions of a class? Explain with examples.
58. What are friend functions? What are the general characteristics of friend functions? Write a C++ program to calculate the factorial value using a friend function.
59. Write a short note on:
 - i. Member function
 - ii. This pointer
60. Write a program to find out the greatest and the smallest among three numbers using pointers.
61. Write a program for passing and returning an array as an argument to a function.
62. What is the difference between a class and a structure?
63. What is a class and object in C++? Explain with suitable examples.
64. Explain friend functions with suitable examples. Write a program to swap two numbers using a friend function.
65. Explain various member functions in C++.
66. In How many ways can a class be Defined in C++ explain with suitable examples?
67. What is the this pointer in C++? Explain with a suitable example.
68. Explain an array in C++. How is memory allocated in an array, explain?
69. How do you declare an array of objects in C++?
70. Write with examples how will you dynamically initialize objects?

71. What is a constructor? Explain types of constructors with examples.
72. What is a copy constructor? Explain with a program.
73. Difference between a constructor and a destructor? Explain a default constructor with examples.
74. What is a default constructor and a parameterized constructor?
75. What is the this pointer (->)? Explain with an example and program.
76. Write the syntax of this pointer (->)? Explain constructors using this pointer.
77. List out the rules for defining constructors with appropriate examples.
78. Write a program of prime number using constructors.
79. Why is the base class constructor called on creating an object of a derived class?
80. How is a constructor different from a normal member function? Write a program of a parameterized constructor.
81. What is inheritance? Explain types of inheritance with a neat diagram.
82. Write a program of inheritance? Explain hierarchical inheritance with examples.
83. What is the difference between public and private access specifiers?
84. Write a program in C++ of inheritance showing that class A is the base class and class B and class C are derived classes.
85. Explain constructor and destructor in multiple inheritance.
86. Explain single inheritance in brief. Write a program of single inheritance.
87. Difference between single inheritance and multiple inheritance with a program.
88. Difference between hierarchical inheritance and hybrid inheritance with a program.
89. What are the advantages of inheritance? Explain in brief.
90. What are the base class and derived class? How to implement inheritance.
91. Write a C++ program to overload unary operators that are increment and decrement.
92. Write a C++ program to overload the binary operator + to add two complex numbers.
93. What is a virtual function? Why do we need virtual functions? Explain with suitable examples.
94. What is operator overloading? How to Define Operator Overloading? Explain with suitable examples.
95. What is the difference between opening a file with constructor function and opening a file with open() function?
96. What is the file access mode? Describe the various file modes.
97. Compare & contrast read(), readline(), and readlines().
98. How many file modes can be used with the open() function to open a file?
99. State the function of each mode.
100. Explain the use of output functions write() and writeline with an example each.
101. Describe the various classes available for file operations.
102. What is the difference between text mode and binary mode in file handling in C++?
103. How do you read data from a file and write data to a file in C++?
104. What is the difference between getline and get functions in C++ when reading from a file?
105. Explain guidelines for exception handling.
106. Explain the mechanism of exception handling.
107. Describe the role of keywords try, catch, and throw in exception handling.

108. How many different types of errors are encountered in a program?
109. Explain specifying exceptions with necessary steps.
110. How can you manage runtime errors? Explain.
111. What are the differences between synchronous and asynchronous exceptions?
112. Explain the exception handling model of C++ with various constructs supported by it.
113. What are the advantages of using the exception handling mechanism in a program?
114. What is a class template? Write the syntax for a class template. Write an example program for a class template.
115. Write a function template to perform a linear search in an array.
116. Write a function template to find the minimum and maximum values by passing non-type arguments to the template.