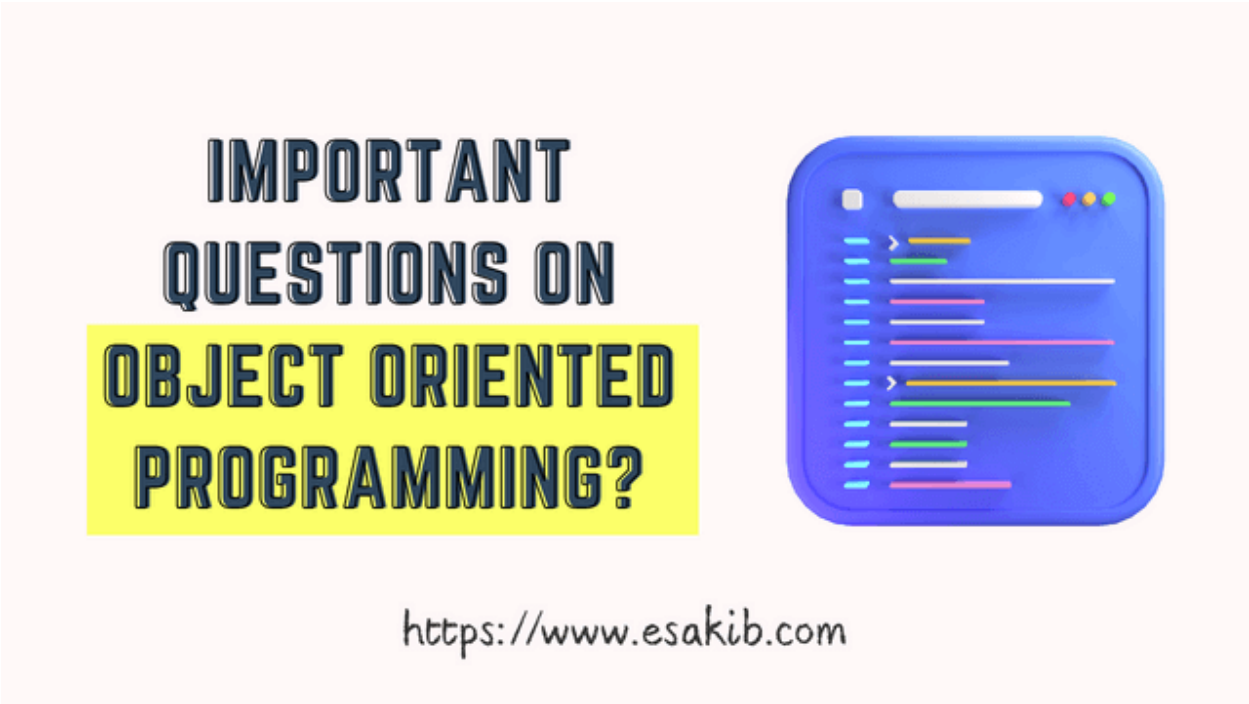

Index

Principles of Object Oriented Programming (OOP) 2

Tokens, Expression, and Control Structure 2

Function (C++/Java) 3

Class and Objects 4

CONSTRUCTORS and DESTRUCTORS 5

Operator Overloading and Type Conversions 5

Inheritance 6

Pointers, Virtual Functions, Polymorphism 6

Console, File, Exception Handling 7

OOP Questions Based on Java 7

Principles of Object Oriented Programming (OOP)

1. What do you mean by object-oriented programming?
2. What do you mean by procedure-oriented programming?
3. State the Characteristics of Procedural oriented programming
4. List the Drawback of Procedural oriented programming.
5. Describe the structure of procedure-oriented programming.
6. What the striking features are of object-oriented programming?
7. Describe the main features of object-oriented programming.
8. Briefly describe the features of object-oriented programming
9. List a few areas of application of OOP technology.
10. Distinguish between procedural-oriented programming and object-oriented programming.
11. Draw the basic organization of data and functions in an object-oriented program. Or. Discuss the organization of data and function in OPP. Or, How data and functions are organized in object-oriented programming?
12. How does the object-oriented approach differ from the object-based approach?
13. What do you mean by dynamic binding? How is it useful in object-oriented programming?
14. Explain the concept of early and late binding.
15. Differentiate early binding with late binding.
16. Write the difference between the following terms: a. Data abstraction and data encapsulation: b. Objects and classes.
17. Define the following terms: Inheritance Data abstraction Data encapsulation.
18. "Encapsulation reduces the complexity"- justify your position.
19. What is C++? 20. Describe the significant parts of a C++ program.
20. What are the Applications of C++?
21. Explain the "iostream" file.
22. What is reusability in C++?
23. Explain the keyword "pure".
24. Does encapsulation reduce the complexity?

Tokens, Expression, and Control Structure

1. Define token.
2. Define Keywords.
3. Define identifier
4. Define constant
5. Define string
6. Define operators
7. Define expression.
8. Explain the basic data types in C++. Or. Explain the data types of C++
9. What do you mean by dynamic initialization of a variable? Give an example.

10. List the different types of operators used in C++ language and state their purposes.
11. What is the role of the scope resolution operator? Explain with an example.
12. What is the new operator?
13. What are the advantages of using a new operator as compared to function malloc()?
14. What is dynamic memory allocation?
15. How can memory be allocated using 'new' and released it using 'delete'?
16. Write down the name of two memory management operators.
17. How can you differentiate the 'new' and 'delete' operator with the malloc() and free() function on behalf of dynamic memory allocation?
18. Describe the memory management operators, introduced in C++.
19. Compare and contrast dynamic memory allocation and deallocation operators new. new [], delete, delete [ ].
20. With examples explain the use of unary and ternary operators.
21. Define the keyword "static". using this keyword write a program in of the number of objects that have been created C++ that keeps count
22. Why is an array called a derived data type?
23. How does a constant defined by const differ from the constant defined by the preprocessor statement #define?
24. What is the bool data type?
25. What is abstract data type?

Function (C++/Java)

1. Define function.
2. What is a friend function?
3. Write down the advantages and disadvantages of using the friend function.
4. When do we declare a function as a friend function?
5. What are the merits of friend function?
6. What are the drawbacks of the friend function?
7. Can a friend function be friends with more than one class?
8. Show with a suitable code segment how the friend function can be defined.
9. A friend function cannot be used to overload the "=" operator. Explain way.
10. What happens when the friend function is used in operator overloading"
11. When a friend function is compulsory? Explain with an example.
12. What is Friend Class?
13. What is an inline function? Or, Explain about inline function.
14. Write down the advantages and disadvantages of the inline function.
15. Write down the situation where inline functions cannot work.
16. How does an in-line function differ from a preprocessor macro?
17. Write a functioning power () to raise a number m to a power n. the function takes a double form and int value for n and returns the result correctly.
18. Explain normal function.
19. Describe the different styles of function prototyping.

20. What is a default argument? Why do we need to use default arguments? Or, What is meant by default argument?
21. Describe ambiguity in the case of default argument with an example.
22. What is meant by volatile function?
23. What do you mean by overloading a function? When do we use overloading of a function concept? Describe with an example.
24. What do you mean by function overloading and function overriding?
25. Differentiate between function overriding and function overloading.
26. What are the characteristics of a member's functions?
27. What are the advantages of function prototypes in C++?
28. What are operator overloading and operator function? Describe the syntax of an operator function.
29. How do ambiguous situations arise from function overloading? Explain with an appropriate example.
30. What are the characteristics of member function?
31. What are the advantages of a function prototype?
32. When a friend is compulsory?
33. How a function can be expended inline?
34. What is meant by default arguments & volatile functions?
35. Define static data member. Mention the properties of static member functions.

Class and Objects

1. What are objects? How are they created?
2. How is memory allocated for objects?
3. Describe memory allocation for objects.
4. How are the objects created in memory?
5. Describe memory allocation for objects
6. In what sequence the constructor and destructor functions are called when you create in C++? more than one object? Explain with an example using both global and local objects.
7. What is the class like? How does it accomplish data hiding?
8. What is a virtual base class?
9. When do we make a class virtual? Discuss with example.
10. What is a generic class and what is its general form?
11. What is a local class? Explain with an example.
12. When do you declare a member of a class static? Explain with an example.
13. Describe the mechanism of accessing data member and member function in the following i) Inside the main program; ii) Inside a member function of the same class.
14. Explain for example, how an object of a class contains objects of other classes created.
15. Write a class to represent the time that includes the member function to perform the following: Take the input for a time in hours and minutes. Multiply time by a scalar value. Add two times. Display the time in the form of hours: or minutes.
16. How does access to the private member function in a class?

17. How is a member function of a class defined?
18. What is a generic class?
19. What is the local class?

CONSTRUCTORS and DESTRUCTORS

1. What is a contractor?
2. Write down the main characteristics of the constructor function.
3. What is a constructor? How is the constructor overloaded?
4. Give two reasons why you need to overload a class's constructor.
5. What do you mean by default constructor and default argument constructor?
6. Briefly describe copy constructor. Or, What is a copy constructor? Show the use of copy constructor with an example.
7. What do you understand by constructor and destructor? Or, What is a constructor? What is a destructor? When are they executed?
8. What are constructors and destructors?
9. Write down the properties of the constructor and destructor.
10. What are the differences between constructors and destructors?
11. Is it mandatory to use constructors in a class?
12. Write a simple program using the copy constructor.
13. In what sequence the constructor and destructor functions are called when you create more than one object? Explain with an example using both global and local objects.
14. Distinguish between default constructor and default value constructor. When do you need the default value constructor?
15. A class complex has a constructor, complex (double r, int i). Can you use this constructor to convert types? Show with example.
16. What is the constructor and Write down the properties of those. destructor functions?
17. Write a C++ program to copy the contents of one file to another.

Operator Overloading and Type Conversions

1. What is operator overloading?
2. What do you mean by function overloading?
3. State the rules for operator overloading.
4. What are the rules of binary operator overloading and unary operator overloading?
5. What is the difference between function overloading and operator overloading? When an operator is overloaded, does it lose any of its original functionality?
6. A friend function cannot be used to overload the assignment operator=. Explain why? Y and we want
7. We have two classes X and Y. If a is an object of X and b is an object of to say a = b; What type of conversion routine should be used and where?
8. Which operators cannot be overloaded in C++and why?

Inheritance

1. What is reusability? How do you achieve this in C++?
2. What does inheritance mean in C++?
3. Write down the importance of inheritance.
4. Write down the Disadvantages of inheritance.
5. What are the different forms of inheritance? Give an example for each.
6. Describe the visibility chart for inheritance. Or, Give a visibility chart for inheritance.
7. When do we use the protected visibility specifier for a class member? Or, What is protected used for?
8. How can we access the class members? Explain with an example.
9. Discuss multilevel inheritance and multiple inheritances with the necessary
10. Write a C++ program showing a multilevel inheritance figure.
11. Describe the syntax of multiple inheritances.
12. What do you mean by multiple inheritances? Explain with a simple program. Or. When do we use multiple inheritances? Explain with an example.
13. Describe the multipath inheritance.
14. What is a virtual base class? When do you make a class virtual? OR. When one base class A derives two classes, B and C, and these two classes derive one child class D, then what will happen? Is there any problem arising? Give your solution in this case.
15. What is the difference between private and protected visibility modifiers?
16. Define abstract base class.
17. What is containership? How does it differ from inheritance?
18. We know that a private member of a base class is not inheritable. Is it anyway possible for the objects of a derived class to access the private members of the base class? If yes, how? Remember the base class cannot be modified.
19. Write a C++ program to illustrate hierarchical inheritance.
20. Explain the different visibility of inheritance.
21. Discuss the problem of multilevel inheritance.
22. Write a C++ program to illustrate hierarchical inheritance.
23. Differentiate between public, private, and protected inheritance with example.
24. How can ambiguity problems be handled in multiple inheritances with examples?

Pointers, Virtual Functions, Polymorphism

1. What are Pointers?
2. What does this pointer point to? Or, What is "this" pointer?
3. What are the applications of this pointer? Or, Mention the application of "this" pointer.
4. What do you mean by a polymorphism in C++? Or, What is polymorphism?
5. How is polymorphism achieved at compile time and run time?
6. What is polymorphism mean in C++? How is polymorphism achieved at (i) compile time and (ii) run time?
7. How can differentiate between compile time polymorphism and run time polymorphism?

8. Define compile time polymorphism.
9. How polymorphism is achieved with function overloading?
10. What is a virtual function? Or, What are virtual functions?
11. Why do we need virtual functions? Or, How does a virtual function help to achieve run time polymorphism?
12. State the importance of virtual function. Or. Why virtual function is important? Or, Why do we need virtual function?
13. What is the basic difference between virtual and pure virtual functions?
14. Write the rules for the virtual function.
15. When do we make a virtual function "pure"? What are the implications of making a function a pure virtual function?
16. How does polymorphism promote extensibility?
17. Why do you need RTII? Suggest some cases where we need to use RTTI.

Console, File, Exception Handling

1. What is a stream? Describe briefly the features of the I/O system supported by C++.
2. What are the input and output streams?
3. What are the main purposes of streams in C++?
4. Briefly the features of the I/O stream supported by C++?
5. How a stream can be designed in C++?
6. Why do we use RTII?
7. What is stream and discuss it?
8. What is the role file () function? When do we use this function?
9. Explain the error handling functions for file operation
10. Describe the various approaches by which we can detect the end-of-file condition
11. Both "ios: ate" and "ios app" place the file pointer at the end of the file (when it is opened). Then what is the difference between them?
12. What is the difference between opening a file with the constructor function and the member function open()? When one method is preferred over the other?
13. What is an exception?
14. What is an exception? How is an expectation handled in C++? Or, what is exception handling and how is it treated in C++?
15. Briefly explain how try, catch and throw work together to provide C++ exception handling
16. What form of catch will handle all types of exceptions?
17. What is a template?

OOP Questions Based on Java

1. What is JAVA?
2. What is JVM?
3. Why java is called the platform-independent programming language?
4. What is function overriding and overloading in Java?

5. Distinguish between Java and C++.
6. What is classpath and byte code in JAVA?
7. What are the differences between applet and servlet in JAVA?
8. What is the task of the main method in a JAVA program?
9. Explain public static void main (string args[]) in JAVA.
10. Explain JDK, JRE, and JVM.