

Question Bank

Basic C++ Questions

1. What is C++?

Answer: C++ is a general-purpose, object-oriented programming language developed by Bjarne Stroustrup. It supports procedural, object-oriented, and generic programming.

2. Who developed C++?

Answer: C++ was developed by Bjarne Stroustrup at Bell Labs.

3. What is a class?

Answer: A class is a user-defined data type that combines data members and member functions into a single unit.

4. What is an object?

Answer: An object is an instance of a class. It represents a real-world entity and can access the members of the class.

5. What is OOP?

Answer: OOP stands for Object-Oriented Programming. It is a programming approach based on objects and classes.

6. What are the main features of OOP?

Answer: The main features are:

Encapsulation

Abstraction

Inheritance

Polymorphism

7. What is encapsulation?

Answer: Encapsulation is the process of combining data and functions into a single unit, usually a class, and controlling access to the data.

8. What is abstraction?

Answer: Abstraction means hiding unnecessary implementation details and showing only the essential features to the user.

9. What is polymorphism?

Answer: Polymorphism means one name having multiple forms. In C++, function overloading and operator overloading are examples of compile-time polymorphism.

10. What is inheritance?

Answer: Inheritance is a mechanism in which a new class acquires the properties and behavior of an existing class.

Classes and Objects

11. What is the syntax for creating a class?

```
class ClassName {  
    // members  
};
```

12. How do you create an object?

```
ClassName obj;
```

13. What are access specifiers in C++?

Answer: The three main access specifiers are:

```
public  
private  
protected
```

14. What is a private member?

Answer: A private member can normally be accessed only from within the same class.

15. What is a public member?

Answer: A public member can be accessed from outside the class through an object.

16. What is a protected member?

Answer: A protected member can be accessed within the class and by derived classes.

17. What is the default access specifier in a C++ class?

Answer: private.

18. What is the difference between a class and an object?

Answer: A class is a blueprint or template, while an object is an instance of that class.

Constructors and Destructors

19. What is a constructor?

Answer: A constructor is a special member function that is automatically called when an object is created.

20. What are the types of constructors?

Answer: Common types are:

Default constructor

Parameterized constructor

Copy constructor

21. What is a default constructor?

Answer: A constructor that takes no arguments is called a default constructor.

```
Student() {  
    cout << "Default constructor";  
}
```

22. What is a parameterized constructor?

Answer: A constructor that accepts one or more arguments is called a parameterized constructor.

```
Student(int r) {  
    roll = r;  
}
```

23. What is a copy constructor?

Answer: A copy constructor creates a new object by copying another object of the same class.

```
Student(const Student &s) {  
    roll = s.roll;  
}
```

24. Can constructors be overloaded?

Answer: Yes. Multiple constructors can exist with different parameter lists.

25. What is a destructor?

Answer: A destructor is a special member function that is automatically called when an object is destroyed.

26. What is the syntax of a destructor?

```
~ClassName() {  
    // cleanup  
}
```

27. Can a destructor be overloaded?

Answer: No. A class can have only one destructor.

28. What is the difference between constructor and destructor?

Answer: A constructor initializes an object when it is created, whereas a destructor performs cleanup when the object is destroyed.

Function Overloading

29. What is function overloading?

Answer: Function overloading means defining multiple functions with the same name but different parameter lists.

30. Give an example of function overloading.

```
int add(int a, int b);  
int add(int a, int b, int c);
```

31. Which type of polymorphism is function overloading?

Answer: Compile-time polymorphism.

32. Can functions be overloaded by changing only the return type?

Answer: No. The parameter list must differ.

33. What are the ways to overload a function?

Answer: Functions can be overloaded by changing:

Number of parameters

Type of parameters

Order of parameters

Friend Function

34. What is a friend function?

Answer: A friend function is a non-member function that can access the private and protected members of a class.

35. Which keyword is used for a friend function?

Answer: The friend keyword.

36. Is a friend function a member of the class?

Answer: No.

37. Can a friend function access private members?

Answer: Yes.

38. How is a friend function called?

Answer: It is called like a normal function.

display(obj);

39. Is friendship inherited?

Answer: No. Friendship is not inherited.

Operator Overloading

40. What is operator overloading?

Answer: Operator overloading allows an existing operator to be given a special meaning when used with objects of a class.

41. Give an example of operator overloading.

```
Complex operator+(Complex c) {  
    // addition of objects  
}
```

42. Which keyword is used in operator overloading?

Answer: The keyword operator.

43. Can all operators be overloaded?

Answer: No. Some operators, such as ::, ., .*, and ?::, cannot be overloaded.

44. Can we create a new operator?

Answer: No. Only existing operators can be overloaded.

45. Can operator precedence be changed through overloading?

Answer: No.

46. Can operator associativity be changed?

Answer: No.

Inheritance

47. What is inheritance?

Answer: Inheritance allows a derived class to acquire the properties and functions of a base class.

48. What is a base class?

Answer: A class whose properties are inherited by another class is called a base class or parent class.

49. What is a derived class?

Answer: A class that inherits properties from another class is called a derived class or child class.

50. What are the types of inheritance?

Answer: Common types are:

Single inheritance

Multiple inheritance

Multilevel inheritance

Hierarchical inheritance

Hybrid inheritance

51. What is single inheritance?

Answer: When one derived class inherits from one base class, it is called single inheritance.

A

|

B

52. What is multiple inheritance?

Answer: When one derived class inherits from more than one base class, it is called multiple inheritance.

A B

\ /

C

53. What is multilevel inheritance?

Answer: When a class is derived from another derived class, it is called multilevel inheritance.

A

|

B

|

C

54. What is hierarchical inheritance?

Answer: When multiple derived classes inherit from one base class, it is called hierarchical inheritance.

A

/\

B C

Polymorphism

55. What is polymorphism?

Answer: Polymorphism means one interface/name having multiple forms.

56. What are the types of polymorphism in C++?

Answer:

Compile-time polymorphism

Run-time polymorphism

57. Give examples of compile-time polymorphism.

Answer: Function overloading and operator overloading.

58. What is run-time polymorphism?

Answer: Run-time polymorphism occurs when the function to be executed is determined during program execution, commonly through virtual functions.

59. What is a virtual function?

Answer: A virtual function is a member function declared using the virtual keyword that enables run-time polymorphism.

Pointers and Memory

60. What is a pointer?

Answer: A pointer is a variable that stores the memory address of another variable.

61. What is the address-of operator?

Answer: The & operator is used to obtain the address of a variable.

62. What is the dereference operator?

Answer: The * operator is used to access the value stored at an address held by a pointer.

63. What is dynamic memory allocation?

Answer: Dynamic memory allocation means allocating memory during program execution using new and releasing it using delete.

```
int *p = new int;  
delete p;
```

Important Rapid-Fire Viva Questions

64. What is the extension of a C++ source file?

Answer: .cpp

65. Which statement is used for output in C++?

Answer: cout

66. Which statement is used for input?

Answer: cin

67. Which header file is commonly used for cout and cin?

Answer: <iostream>

68. What does endl do?

Answer: It inserts a new line and flushes the output stream.

69. What is this pointer?

Answer: this is a pointer that refers to the current object.

70. What is a namespace?

Answer: A namespace is used to organize identifiers and avoid naming conflicts. For example:

using namespace std;

71. What is the purpose of public?

Answer: It allows members to be accessed from outside the class.

72. What is the purpose of private?

Answer: It restricts direct access to members from outside the class.

73. What is the purpose of protected?

Answer: It allows access within the class and derived classes.

74. What is a static data member?

Answer: A static data member is shared by all objects of a class rather than having a separate copy for each object.

75. What is the difference between = and ==?

Answer: = is the assignment operator, while == is the equality comparison operator.

76. What is the difference between ++i and i++?

Answer: ++i is pre-increment, where the value is incremented before use. i++ is post-increment, where the value is used first and then incremented.

77. What is the main function in C++?

Answer: main() is the entry point from which execution of a C++ program begins.

78. What does return 0 mean in main()?

Answer: It indicates that the program completed successfully.

79. What is compile-time polymorphism?

Answer: It is polymorphism where the function/operator to be used is determined by the compiler, such as function overloading and operator overloading.

80. What is run-time polymorphism?

Answer: It is polymorphism where the appropriate function implementation is determined at run time, commonly using virtual functions.