1. What is a class?
Ans: The objects with the same data structure (attributes) and behavior (operations) are called class.

2. What is an object?
Ans: It is an entity which may correspond to real-world entities such as students, employees, bank account. It may be concrete such as file system or conceptual such as scheduling policies in multiprocessor operating system.
Every object will have data structures called attributes and behavior called operations.

3. What is the difference between an object and a class?
Ans: All objects possessing similar properties are grouped into class.
Example :–person is a class, ram, hari are objects of person class. All have similar attributes like name, age, sex and similar operations like speak, walk.

```
Class person
{
private:
char name[20];
int age;
char sex;
public: speak();
walk();
};
```

4. What is the difference between class and structure?
Ans: In class the data members by default are private but in structure they are by default public

5. Define object based programming language?
Ans: Object based programming language support encapsulation and object identity without supporting some important features of OOPs language.
Object based language=Encapsulation + object Identity

6. Define object oriented language?
Ans: Object-oriented language incorporates all the features of object based programming languages along with inheritance and polymorphism.
Example: – c++, java.

7. Define OOPs?
Ans: OOP is a method of implementation in which programs are organized as co-operative

collection of objects, each of which represents an instance of some class and whose classes are all member of a hierarchy of classes united through the property of inheritance.

8. What is public, protected, and private?
Ans: These are access specifier or a visibility lebels .The class member that has been declared as private can be accessed only from within the class. Public members can be accessed from outside the class also. Within the class or from the object of a class protected access limit is same as that of private but it plays a prominent role in case of inheritance

9. What is a scope resolution operator?
Ans: The scope resolution operator permits a program to reference an identifier in the global scope that has been hidden by another identifier with the same name in the local scope.

10. What do you mean by inheritance?
Ans: The mechanism of deriving a new class (derived) from an old class (base class) is called inheritance. It allows the extension and reuse of existing code without having to rewrite the code from scratch.

11. What is abstraction?
Ans: The technique of creating user-defined data types, having the properties of built-in data types and a set of permitted operators that are well suited to the application to be programmed is known as data abstraction. Class is a construct for abstract data types (ADT).

12. What is encapsulation?
Ans: It is the mechanism that wraps the data and function it manipulates into single unit and keeps it safe from external interference.

13. How variable declaration in c++ differs that in c?
Ans: C requires all the variables to be declared at the beginning of a scope but in c++ we can declare variables anywhere in the scope. This makes the programmer easier to understand because the variables are declared in the context of their use.

14. What are the c++ tokens?
Ans: c++ has the following tokens
I. keywords
II. Identifiers
III. Constants
IV. Strings
V. operators

15. What do you mean by reference variable in c++?
Ans: A reference variable provides an alias to a previously defined variable.
Data type & reference-name = variable name

16. What do you mean by implicit conversion?
Ans: Whenever data types are mixed in an expression then c++ performs the conversion automatically.
Here smaller type is converted to wider type.
Example- in case of integer and float integer is converted into float type.

17. What is the difference between method overloading and method overriding?
Ans: Overloading a method (or function) in C++ is the ability for functions of the same name to be defined as long as these methods have different signatures (different set of parameters). Method overriding is the ability of the inherited class rewriting the virtual method of the base class.

18. What are the defining traits of an object-oriented language?
The defining traits of an object-oriented language are:
encapsulation
inheritance
polymorphism
Ans:
Polymorphism: is a feature of OOPL that at run time depending upon the type of object the appropriate method is called.

Inheritance: is a feature of OOPL that represents the "is a" relationship between different objects (classes). Say in real life a manager is a employee. So in OOPL manger class is inherited from the employee class.

Encapsulation: is a feature of OOPL that is used to hide the information.

19. What is polymorphism?
Ans: Polymorphism is the idea that a base class can be inherited by several classes. A base class pointer can point to its child class and a base class array can store different child class objects.

20. What do you mean by inline function?
Ans: An inline function is a function that is expanded inline when invoked.ie. the compiler replaces the function call with the corresponding function code. An inline function is a function that is expanded in line when it is invoked. That is the compiler replaces the function call with the corresponding function code (similar to macro).

21 What is the difference between a NULL pointer and a void pointer?
Ans: A NULL pointer is a pointer of any type whose value is zero. A void pointer is a pointer to an object of an unknown type, and is guaranteed to have enough bits to hold a pointer to any object. A void pointer is not guaranteed to have enough bits to point to a function (though in general practice it does).

22. What is difference between C++ and Java?
Ans: C++ has pointers Java does not.
Java is platform independent C++ is not.
Java has garbage collection C++ does not.

23. What do you mean by multiple inheritance in C++ ?
Ans: Multiple inheritance is a feature in C++ by which one class can be of different types. Say class teaching Assistant is inherited from two classes say teacher and Student.

24. What do you mean by virtual methods?
Ans: virtual methods are used to use the polymorphism feature in C++. Say class A is inherited from class B. If we declare say function f() as virtual in class B and override the same function in class A then at runtime appropriate method of the class will be called depending upon the type of the object.

25. What do you mean by static methods?
Ans: By using the static method there is no need creating an object of that class to use that method. We can directly call that method on that class. For example, say class A has static function f(), then we can call f() function as A.f(). There is no need of creating an object of class A.

26. How many ways are there to initialize an int with a constant?
Ans: Two.
There are two formats for initializers in C++ as shown in the example that follows. The first format uses the traditional C notation. The second format uses constructor notation.
int foo = 123;
int bar (123);

27. What is a constructor?
Ans: Constructor is a special member function of a class, which is invoked automatically whenever an instance of the class is created. It has the same name as its class.

28. What is destructor?
Ans: Destructor is a special member function of a class, which is invoked automatically whenever an object goes out of the scope. It has the same name as its class with a tilde character prefixed.

29. What is an explicit constructor?
Ans: A conversion constructor declared with the explicit keyword. The compiler does not use an explicit constructor to implement an implied conversion of types. It's purpose is reserved explicitly for construction.

30 What is the Standard Template Library?
Ans: A library of container templates approved by the ANSI committee for inclusion in the standard C++ specification. A programmer who then launches into a discussion of the generic programming model, iterators, allocators, algorithms, and such, has a higher than average understanding of the new technology that STL brings to C++ programming.

31. What problem does the namespace feature solve?
Ans: Multiple providers of libraries might use common global identifiers causing a name collision when an application tries to link with two or more such libraries. The namespace feature surrounds a library's external declarations with a unique namespace that eliminates the potential for those collisions. This solution assumes that two library vendors don't use the same namespace identifier, of course.

32. What is the use of 'using' declaration?
Ans: A using declaration makes it possible to use a name from a namespace

33. What is a template?
Ans: Templates allow us to create generic functions that admit any data type as parameters and return a value without having to overload the function with all the possible data types. Until certain point they fulfill the functionality of a macro. Its prototype is any of the two following ones:
template function_declaration;
template function_declaration;

34. Differentiate between a template class and class template?
Ans:
Template class:
A generic definition or a parameterized class not instantiated until the client provides the needed information. It's jargon for plain templates.
Class template:
A class template specifies how individual classes can be constructed much like the way a class specifies how individual objects can be constructed. It's jargon for plain classes.

35. What is the difference between a copy constructor and an overloaded assignment operator?
Ans: A copy constructor constructs a new object by using the content of the argument object.

An overloaded assignment operator assigns the contents of an existing object to another existing object of the same class.

36. What is a virtual destructor?
Ans: The simple answer is that a virtual destructor is one that is declared with the virtual attribute.

37. What is an incomplete type?
Ans: Incomplete type refers to pointers in which there is non availability of the implementation of the referenced location or it points to some location whose value is not available for modification.
Example:
int *i=0×400 // i points to address 400
*i=0; //set the value of memory location pointed by i.
Incomplete types are otherwise called uninitialized pointers.

38. What do you mean by Stack unwinding?
Ans: It is a process during exception handling when the destructor is called for all local objects between the place where the exception was thrown and where it is caught.

39. What is a container class? What are the types of container classes?
Ans: A container class is a class that is used to hold objects in memory or external storage. A container class acts as a generic holder. A container class has a predefined behavior and a well-known interface. A container class is a supporting class whose purpose is to hide the topology used for maintaining the list of objects in memory. When a container class contains a group of mixed objects, the container is called a heterogeneous container; when the container is holding a group of objects that are all the same, the container is called a homogeneous container

40. Name some pure object oriented languages?
Ans: Smalltalk, Java, Eiffel, Sather.

41. Name the operators that cannot be overloaded?
Ans: sizeof, ., .*, .->, ::, ?:

42. What is an adaptor class or Wrapper class?
Ans: A class that has no functionality of its own. Its member functions hide the use of a third party software component or an object with the non-compatible interface or a non-object-oriented implementation.

43. What is a Null object?
Ans: It is an object of some class whose purpose is to indicate that a real object of that class

does not exist. One common use for a null object is a return value from a member function that is supposed to return an object with some specified properties but cannot find such an object.

44. What is class invariant?

Ans: A class invariant is a condition that defines all valid states for an object. It is a logical condition to ensure the correct working of a class. Class invariants must hold when an object is created, and they must be preserved under all operations of the class. In particular all class invariants are both preconditions and post-conditions for all operations or member functions of the class.

45. What is a dangling pointer?

Ans: A dangling pointer arises when you use the address of an object after its lifetime is over. This may occur in situations like returning addresses of the automatic variables from a function or using the address of the memory block after it is freed. Example: The following code snippet shows this:

```
class Sample
{
public:
int *ptr;
Sample(int i)
{
ptr = new int(i);
}
~Sample()
{
delete ptr;
}
void PrintVal()
{
cout << "The value is " << *ptr;
}
};
void SomeFunc(Sample x)
{
cout << "Say i am in someFunc " << endl;
}
int main()
{
Sample s1= 10;
SomeFunc(s1);
s1.PrintVal();
}
```

In the above example when PrintVal() function is called it is called by the pointer that has been freed by the destructor in SomeFunc.

46. Differentiate between the message and method?
Ans:
Message:
Objects communicate by sending messages to each other.
A message is sent to invoke a method.
Method
Provides response to a message and it is an implementation of an operation

47. How can we access protected and private members of a class?
Ans: In the case of members protected and private, these could not be accessed from outside the same class at which they are declared. This rule can be transgressed with the use of the friend keyword in a class, so we can allow an external function to gain access to the protected and private members of a class.

48. Can you handle exception in C++?
Ans: Yes we can handle exception in C++ using keyword: try, catch and throw. Program statements that we want to monitor for exceptions are contained in a try block. If an exception occurs within the try block, it is thrown (using throw).The exception is caught, using catch, and processed.

49. What is virtual function?
Ans: A virtual function is a member function that is declared within a base class and redefined by a derived class .To create a virtual function, the function declaration in the base class is preceded by the keyword virtual.

50. What do you mean by early binding?
Ans: Early binding refers to the events that occur at compile time. Early binding occurs when all information needed to call a function is known at compile time. Examples of early binding include normal function calls, overloaded function calls, and overloaded operators. The advantage of early binding is efficiency.

51. What do you mean by late binding?
Ans: Late binding refers to function calls that are not resolved until run time. Virtual functions are used to achieve late binding. When access is via a base pointer or reference, the virtual function actually called is determined by the type of object pointed to by the pointer.