OOPS

QN1.

In object-oriented programming, a class is a template definition of the methods and variables in a particular kind of object.

QN2.

In object-oriented programming (OOP), objects are the things you think about first in designing a program and they are also the units of code that are eventually derived from the process.

QN3.

|  |  |
| --- | --- |
| CLASS | INTERFACE |
| **-Keyword used** | -**Keyword not used** |
| -**Objects can be created** | -**Objects cannot be created** |
| -**It can inherit another class** | -**It cannot inherit another class** |
| -Does not a**llow you to contain abstract methods** | -**Allows you to contain abstract methods** |

QN4.

Method overloading is a form of polymorphism in OOP. Polymorphism allows objects or methods to act in different ways, according to the means in which they are used.

QN5.

Data hiding is an object-oriented programming (OOP) technique specifically used to hide internal object details (i.e., data members).

QN6.

1. Definition: An abstract class is a class that cannot be instantiated and can have both abstract and non-abstract methods (i.e., regular methods with implementations). An interface, on the other hand, is a collection of abstract methods (i.e., methods without implementations) and constants that can be implemented by multiple classes.
2. Multiple inheritance: A class can inherit from only one abstract class, but it can implement multiple interfaces. This means that a class can extend one abstract class and implement multiple interfaces at the same time.
3. Method implementation: In an abstract class, you can provide implementations for some or all of its methods, including regular methods. In an interface, all methods are abstract and do not have any implementation. Classes that implement an interface must provide implementations for all its methods.
4. Fields and constants: An abstract class can have instance variables (fields) and constants, while an interface can only have constants (i.e., static final fields).

QN7.

A virtual function is a member function that you expect to be redefined in derived classes. When you refer to a derived class object using a pointer or a reference to the base class, you can call a virtual function for that object and execute the derived class's version of the function.

QN8.

In class-based, object-oriented programming, a constructor is a special type of subroutine called to create an object. It prepares the new object for use, often accepting arguments that the constructor uses to set required member variables.

QN 9.

In object-oriented programming, abstraction is one of three central principles (along with encapsulation and inheritance). Through the process of abstraction, a programmer hides all but the relevant data about an object in order to reduce complexity and increase efficiency.

QN 10.

Final keyword in C++ when added to a function, prevents it from being overridden by derived classes. Also when added to a class prevents inheritance of any type.

QN11.

A pure virtual function in c++ is defined as a function that is only initialized but not defined. A pure virtual function in c++ must end with " =0 " when declared. This type of function is a concept of Run-time Polymorphism. This type of function needs to be redefined in the derived class.

QN12.

sealed is a context-sensitive keyword for ref classes that indicates that a virtual member cannot be overridden, or that a type cannot be used as a base type. The ISO C++11 Standard language introduced the final keyword. Use final on standard classes, and sealed on ref classes.

QN13.

Runtime polymorphism, also known as the Dynamic Method Dispatch, is a process that resolves a call to an overridden method at runtime. The process involves the use of the reference variable of a superclass to call for an overridden method.

QN14.

 In object-oriented languages, access modifiers (or access specifiers) are keywords that control the accessibility of classes, methods, and other members. Access modifiers are a type of programming language syntax that makes it easier to encapsulate components.

QN15.

In object-oriented programming, a friend function, that is a "friend" of a given class, is a function that is given the same access as methods to private and protected data. A friend function is declared by the class that is granting access, so friend functions are part of the class interface, like methods.

QN16.

If derived class defines same function as defined in its base class, it is known as function overriding in C++. It is used to achieve runtime polymorphism. It enables you to provide specific implementation of the function which is already provided by its base class.

QN17.

The mutable storage class specifier is used only on a class data member to make it modifiable even though the member is part of an object declared as const . You cannot use the mutable specifier with names declared as static or const , or reference members. the compiler would not allow the assignment var2.