**1. What are classes?**A class is a collection of objects of similar type. Once a class is defined any number of objects can be created of that class.
Examples of classes are class of cars, class of pens, class of birds etc.

**What are Instances?**

Instances are essentially objects derived out of classes.

When an object is created for a particular class the process is called Instantiation. In general, life time of object is nothing but the time between an object creation, till the object is no longer used and is destructed or freed.

**2. What is object orientation?**It is a technique for system modeling. It offers a number of concepts

**3. Pointer type:**A Pointer is a variable which holds the address of another variable.

**4. Const Qualifier:** Const qualifier is used to prevent from accidental changes within a program.

**5. Differences between Structured Programming and OOP**

– In St.P emphasis is on What is happening. In OOP emphasis is on Who is being affected.
– Data move openly around the system. In St.P. In OOP the data and functions that operate on the data are tied together
– In St.P most fns share global data. In OOP data is hidden and cannot be accessed by external functions.
– In St.P, Large programs are divided to smaller programs known as functions, In OOP they are divided into objects.

**6. Enumerated Data type**: It is a user-defined data type.
• The syntax of enum statement is similar to that of struct
• For Ex: enum shape{ circle, square, triangle};

**7. New and Delete:** (Memory Management Operators)

These are the unary operators to perform dynamic memory allocation and deallocation
New operator is used to create objects. The General Form is:
Pointer-variable = new data-type; The general form of using delete
delete pointer-variable;

For Ex: delete p;
To free dynamically allocated array
delete [size] pointer-variable
For Ex: delete [10] p;
For Ex: int \*p = new int;

**8. Inline Functions:**

An inline function is a function that is expanded in line when it is invoked

Inline function-header
{function body
}Ex: inline int square(int a)
{return a\*a
};

**9. Generic Functions**

A generic function defines a general set of operations that can be applied to various types of data.. A generic function is created using the keyword template. The general form of template function definition is

1. Template ret-type func-name(parameter list)
{ //function body }

**10. Constructors and Destructors**

1. C++ allows automatic initialization of objects when they are created. This is performed through the use of a constructor function.
2. Constructor is a special function that is a member of the class and has the same name as that class.
3. The Constructor is automatically called whenever an object of its associated class is created.
4. A Constructor is declared and defined as follows

**Destructors:**

1. A destructor as the name implies is a complement of the constructor, used to destroy objects.
2. It will be invoked implicitly by the compiler upon exit from program or function or block
3. Local objects are destroyed when the block is left. Global objects are destroyed when the program terminates.
4. It is a member function whose name is same as the class name but is preceded by a tilde (~) operator. Ex : ~stack();
5. There are many reasons why a destructor may be needed. Ex: An object may need to deallocate memory that it had previously allocated, or close a file that it had opened.
6. A destructor never takes any argument nor does it return any value
7. It is a good practice to declare destructors

**11. Class is syntactically similar to a struct**. The only difference between them is that by default all members are public in a struct and private in a class.

**12. Friend Functions**

1. Private members cannot be accessed from outside the class i.e by a non-member function
2. C++ allows a non-member function to access private members of a class by using a friend function.
3. A friend function need not be a member of any class.
4. To make an outside function friendly to a class, include its prototype within the class, preceding it with the keyword friend.

**13. Static Data Members(SDM)**

1. Preceding a data member’s declaration with the keyword static tells the compiler that only one copy of that variable will exist and that all objects of the class will share that variable.
2. A static data member is initialized to zero when the first object of its class is created. No other initialization is permitted.
3. Declaring a static data member is not defining it (means not allocating storage for it). Remember class is a logical construct that does not have physical reality.
4. Defining a static data member is done outside the class by redeclaring the static variable using the scope resolution operator.
5. SDM are normally used to maintain values common to entire class Static Member Functions(SMF)

There are several restrictions placed on Static member functions.

1. A static function can have access to only other static members declared in the same class.
2. Global functions and data may be accessed by SMF.
3. A SMF does not have a this pointer.
4. There cannot be a static version and non-static version of the same function
5. A SMF may not be virtual.
6. They cannot be declared as const or volatile
7. A static member function can be called using the class name as follows Classname :: function-name;
8. Use is to preinitialize private static data before any object is created

**14. This Pointer**When a member function is called, it is automatically passed an implicit argument that is a pointer to the invoking object. This pointer is called this

**15.Operator Overloading**The mechanism of giving additional meaning to an operator is known as operator overloading.

**16. INHERITANCE**

1. The mechanism of deriving a new class from an old one is called Inheritance.
2. The concept of Inheritance provides the idea of reusability. This is basically done by creating new classes, reusing the properties of the existing ones.
3. A class that is inherited is referred to as base class, and a class that inherits is called the derived class.
4. Different types of Inheritance:

**17. Protected Members**When a member of a class is declared as protected, that member is not accessible by other nonmember elements of the program

**18. 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, precede the function’s declaration in the base class with the keyword Virtual.

**19. I/O Operations:** C++ uses the concept of stream and stream classes to implement the I/O Operations. A stream acts as an interface between the program and I/O device.

**20. What is Polymorphism?**It is generally ability to appear in many forms. In OOP, polymorphism refers to a programming language’s ability to process objects differently depending on their data type or class.

**21 What is Abstraction?**The process of picking out (abstracting) common features of objects and procedures. A programmer would use abstraction. Abstraction is one of the most important techniques in software engineering and it is closely related to two other techniques- encapsulation and information hiding. All these three techniques are used to reduce complexity.