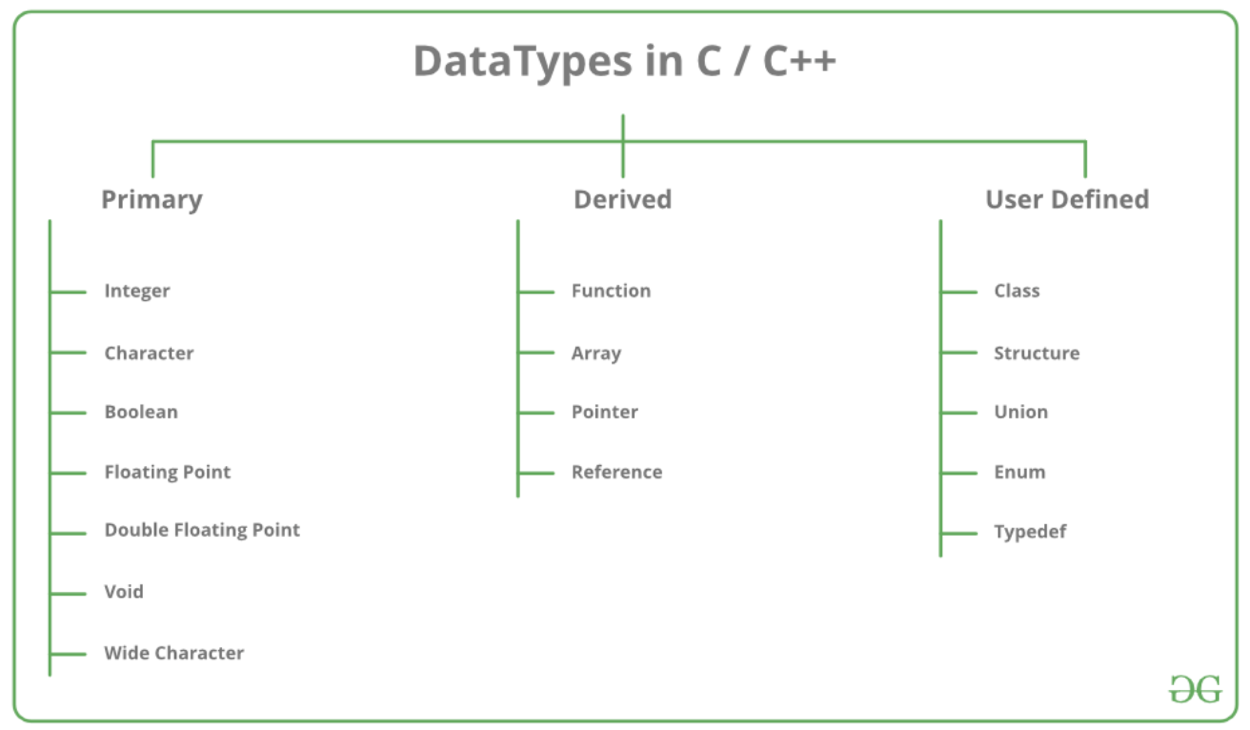
Questions and Answers

**Q1. What are operators?**

**Ans.** An Operators operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. C++ is rich in built-in operators and provide operators like Arithmetic, Relational, Logical, Bitwise, Assignment operator.

**Q2. What are the different data types present in C++?**

**Ans.**

**Q3.  Define ‘std’**

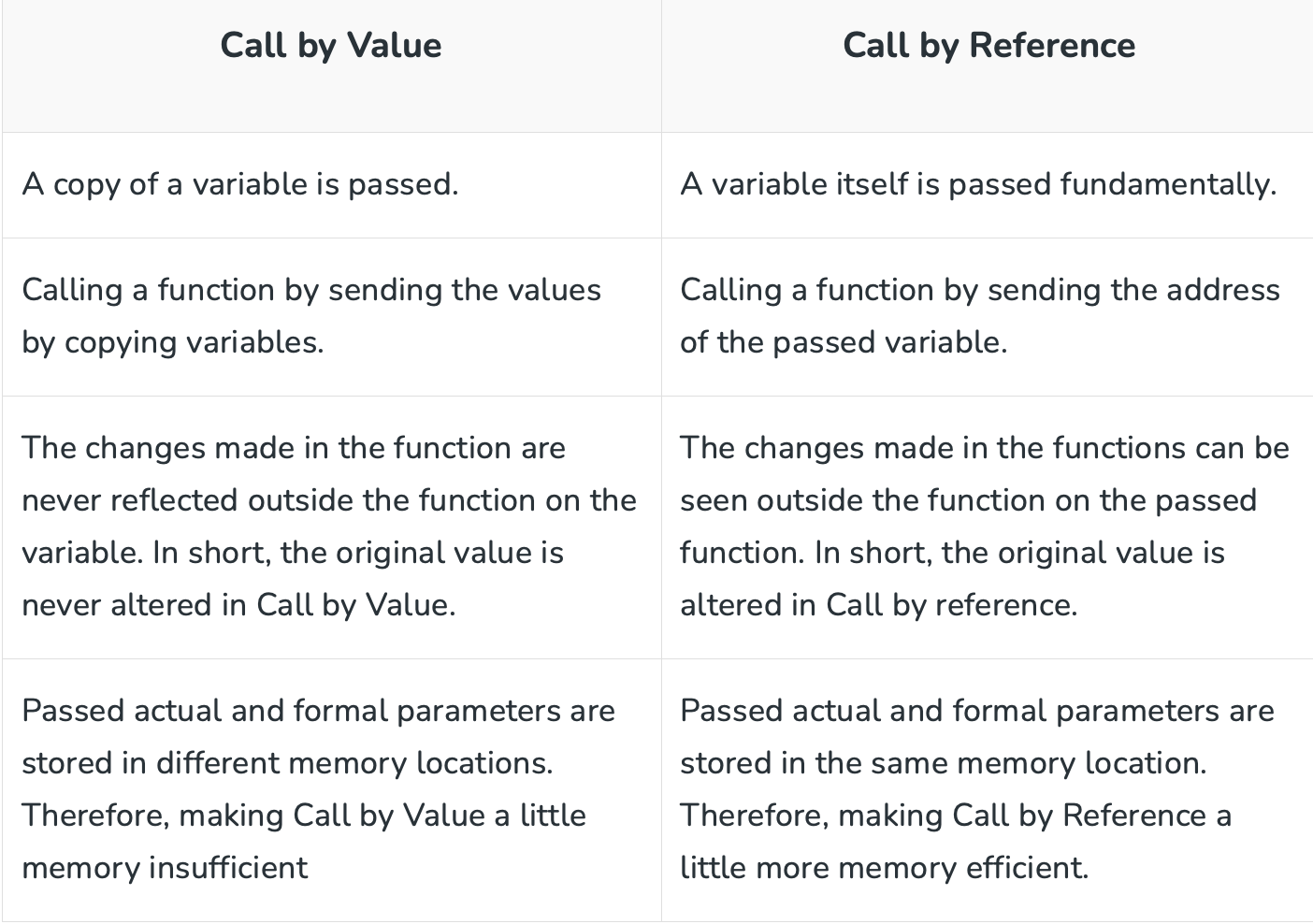
**Ans.** ‘std’ is also known as Standard or it can be interpreted as a namespace. The command “*using namespace std”*informs the compiler to add everything under the *std namespace* and inculcate them in the *global namespace*. This all inculcation of global namespace benefits us to use “cout” and “cin” without using “std::\_operator\_”.

**Q4. What are references in C++?**

**Ans.**

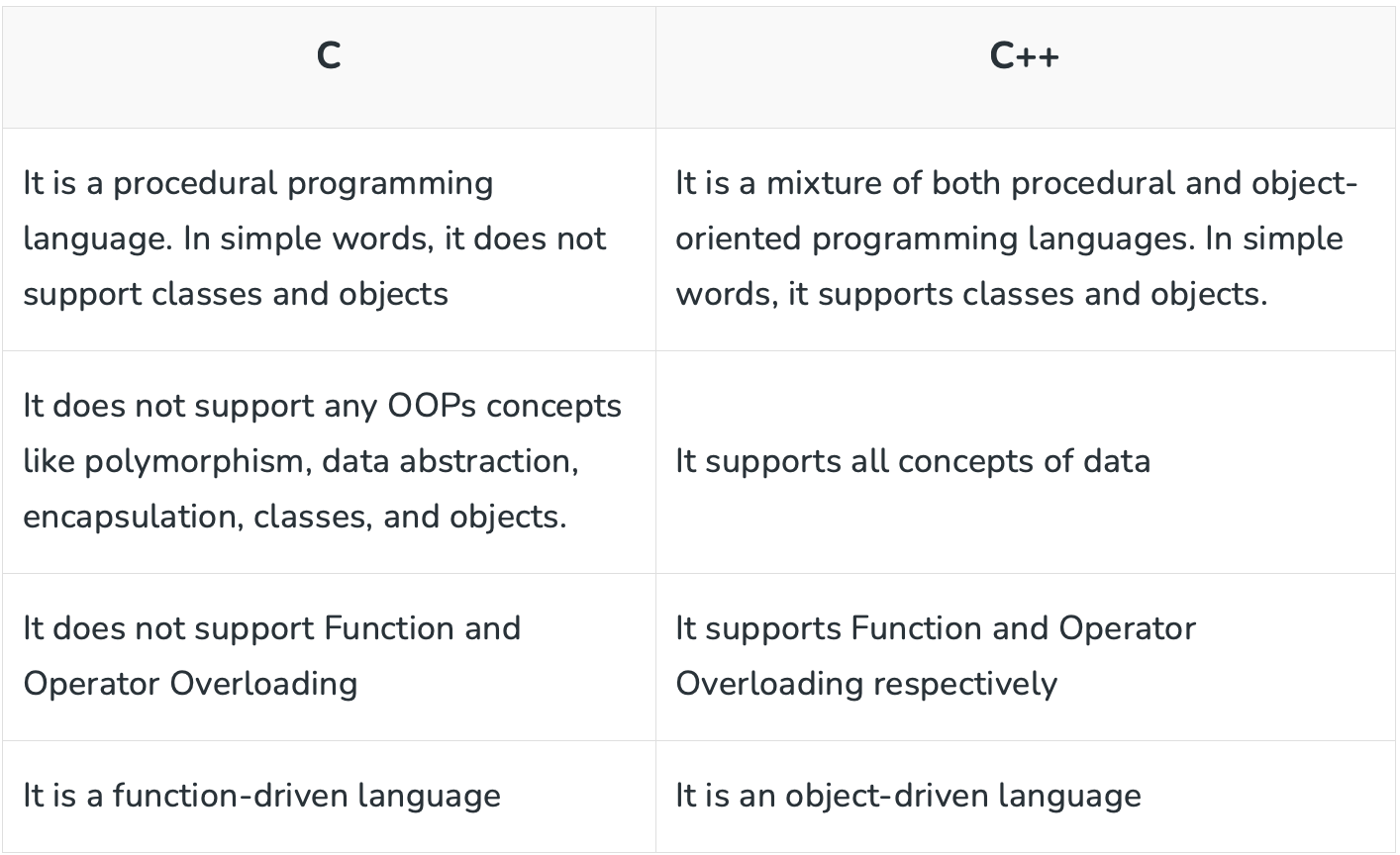
When a variable is described as a reference it becomes an alias of the already existing variable. Keeping in mind that changes made in the reference variable will be reflected in the already existing variable. A reference variable is preceded with a ‘&’ symbol.

*int& ref = GFG;*

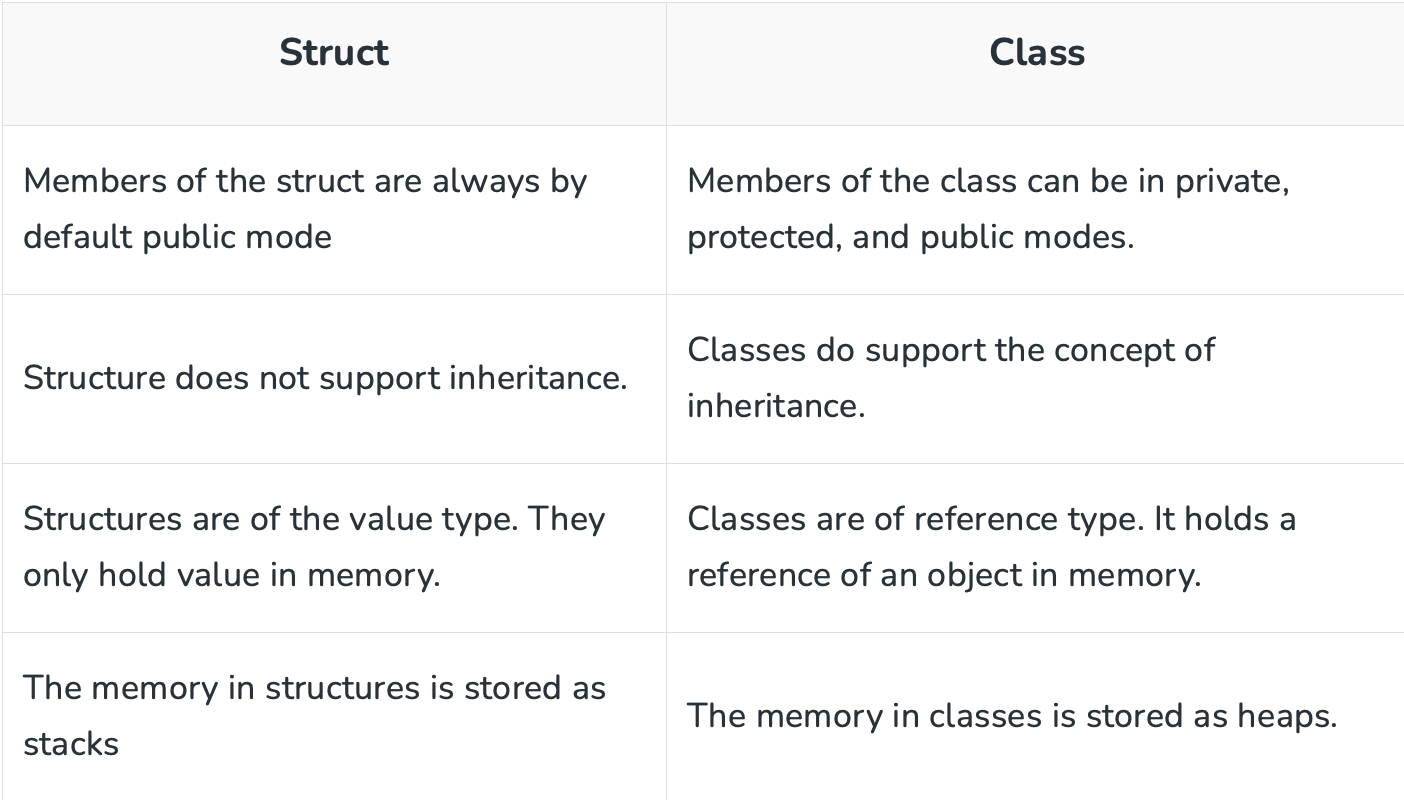
**Q6. What do you mean by Call by Value and Call by Reference?**

**Ans.**

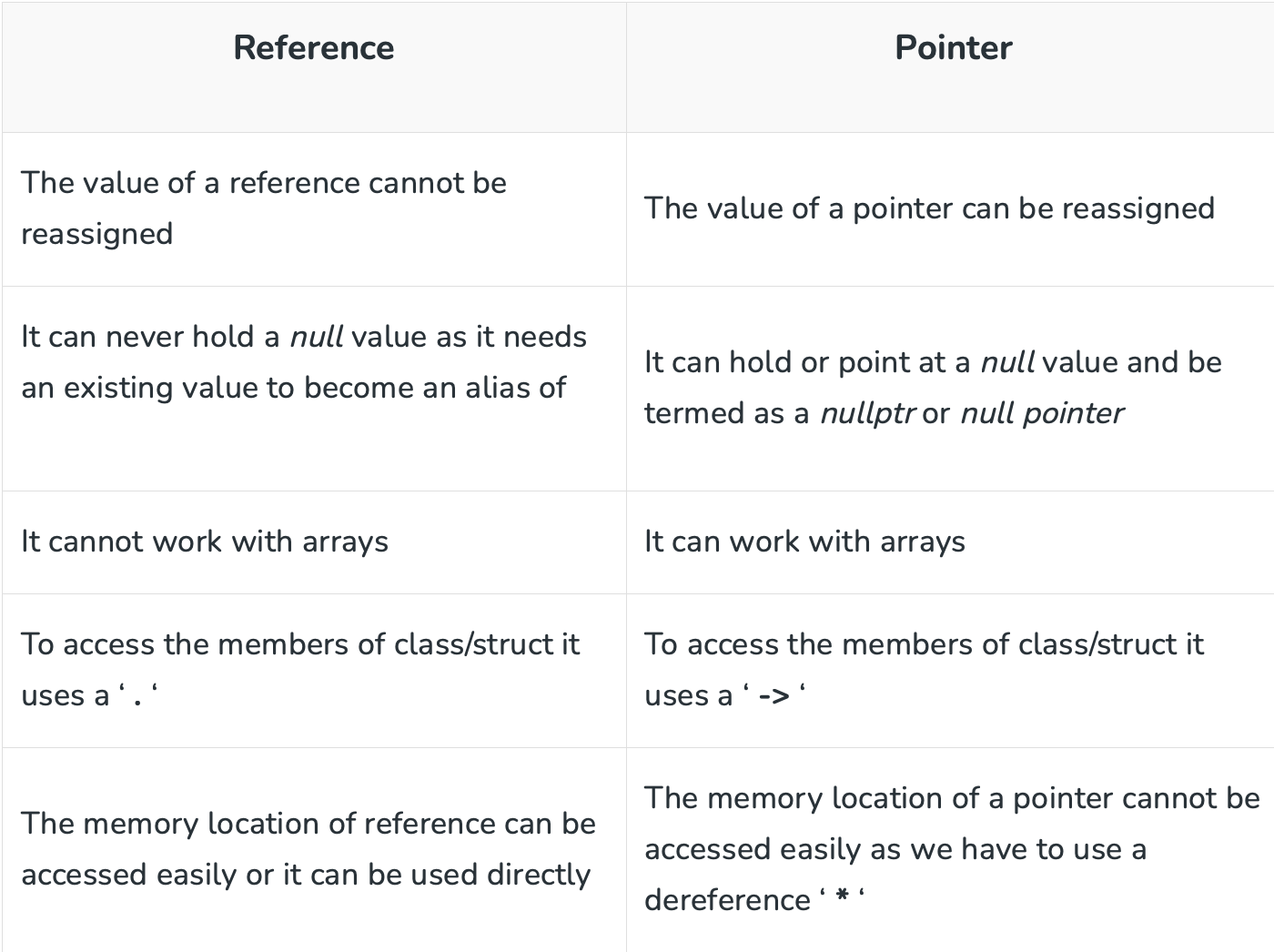
**Q7 What is the difference between C and C++?**

**Ans.**

**Q8. What is the difference between struct and class?**

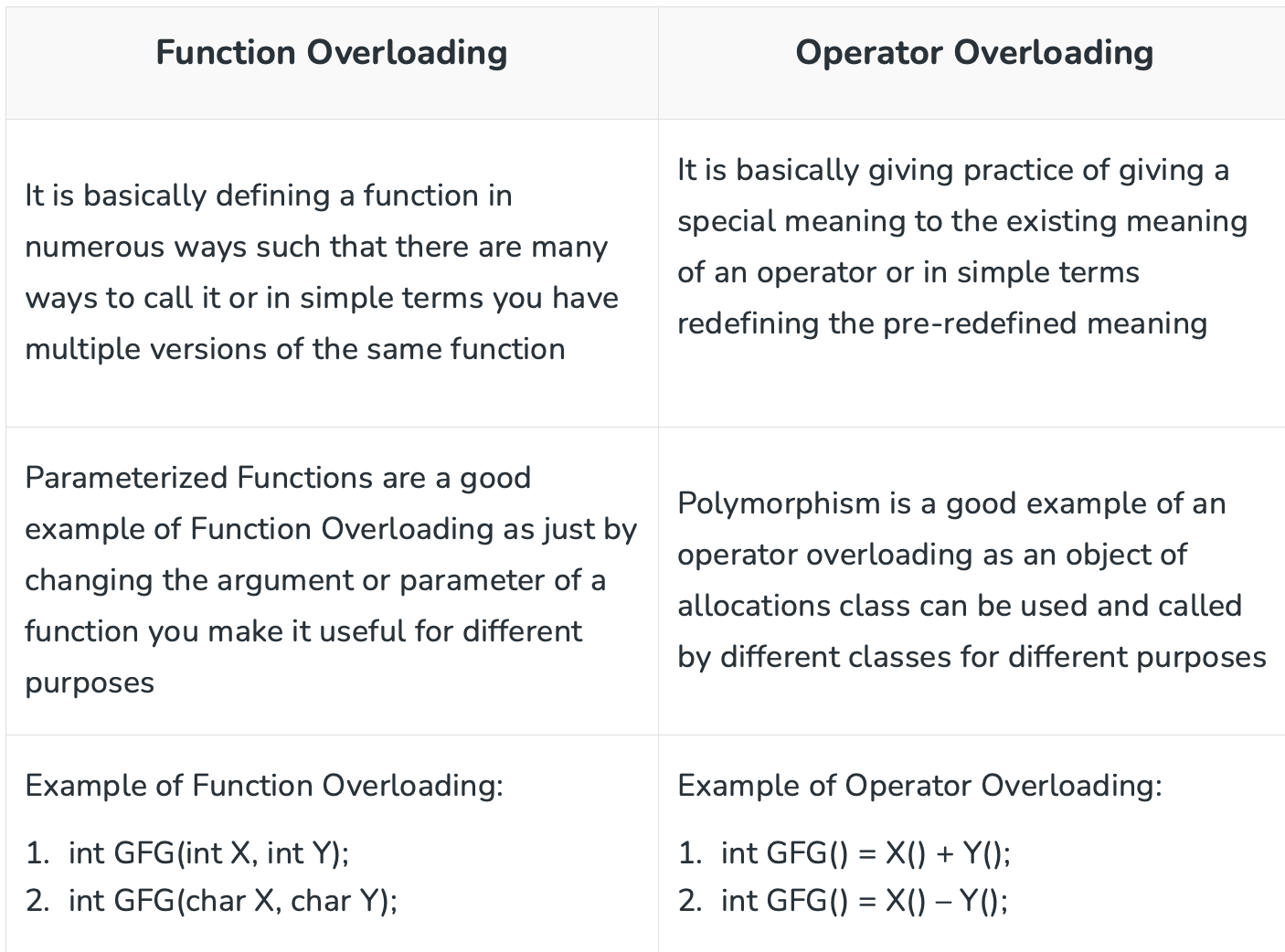
**Ans.**

**Q9. What is the difference between Reference and Pointers?**

**Ans.**

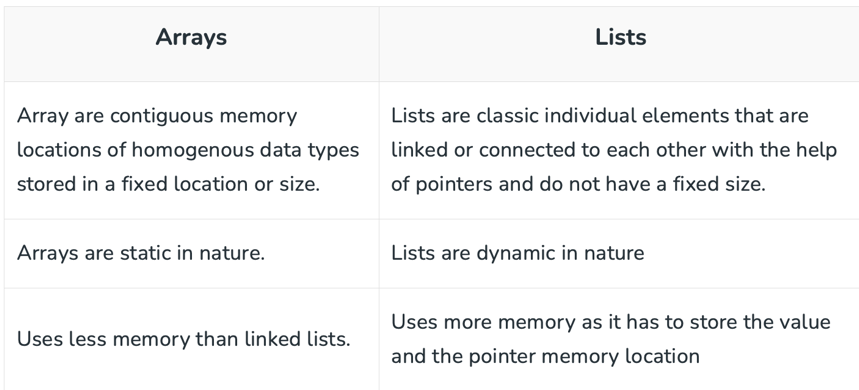
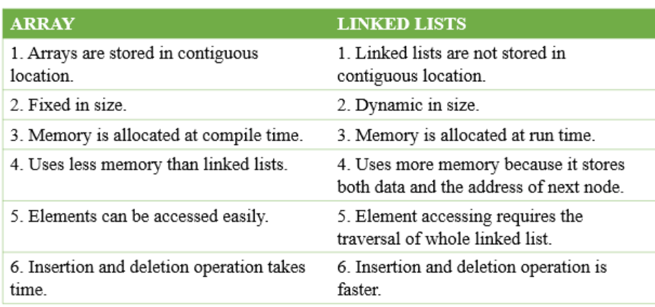
**Q10.** **What is the difference between function overloading and operator overloading?**

**Ans.**

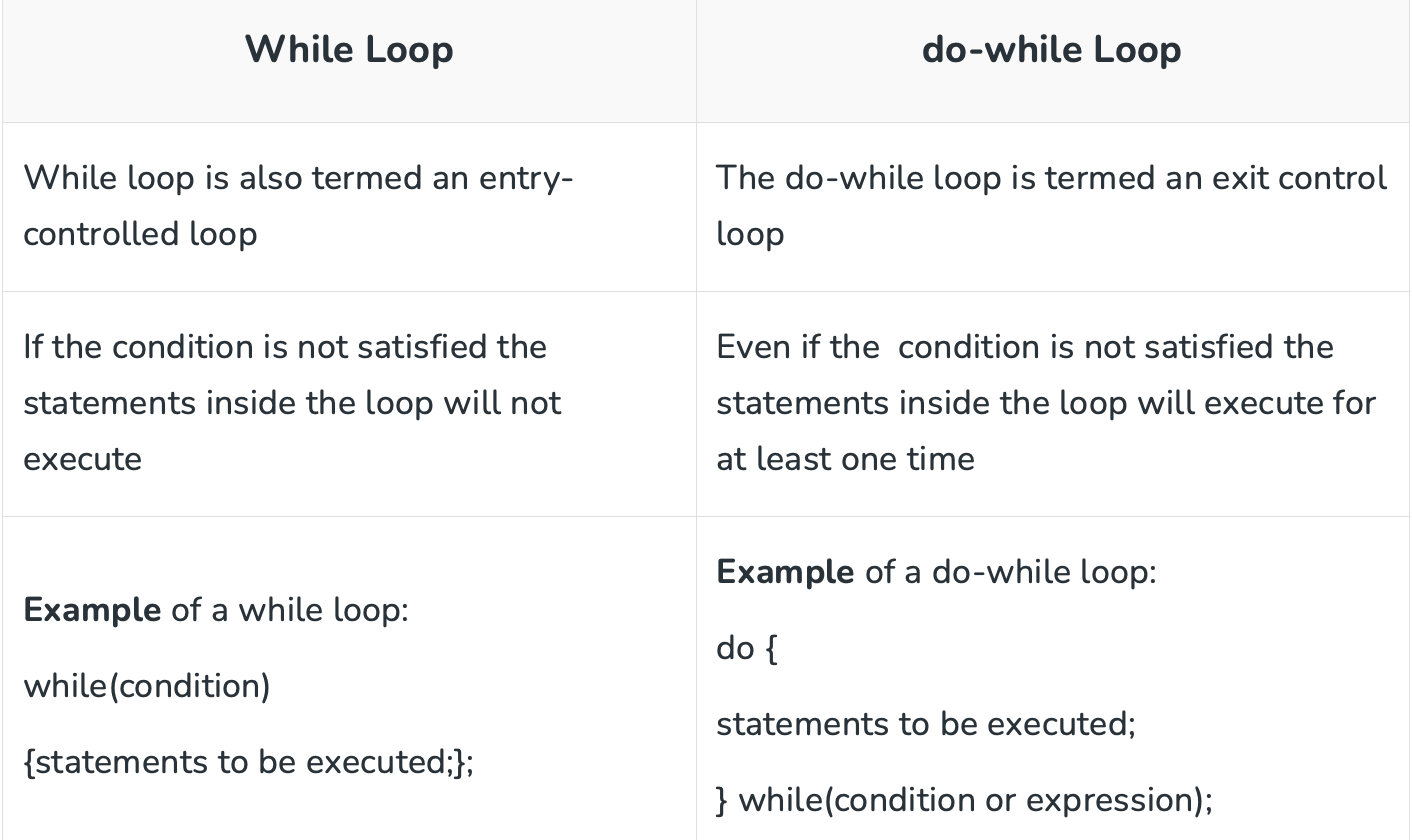
****

\*\* Which operator cannot be overloaded in C++? ::

**Q11. What is the difference between an array and a list?**

**Ans.**

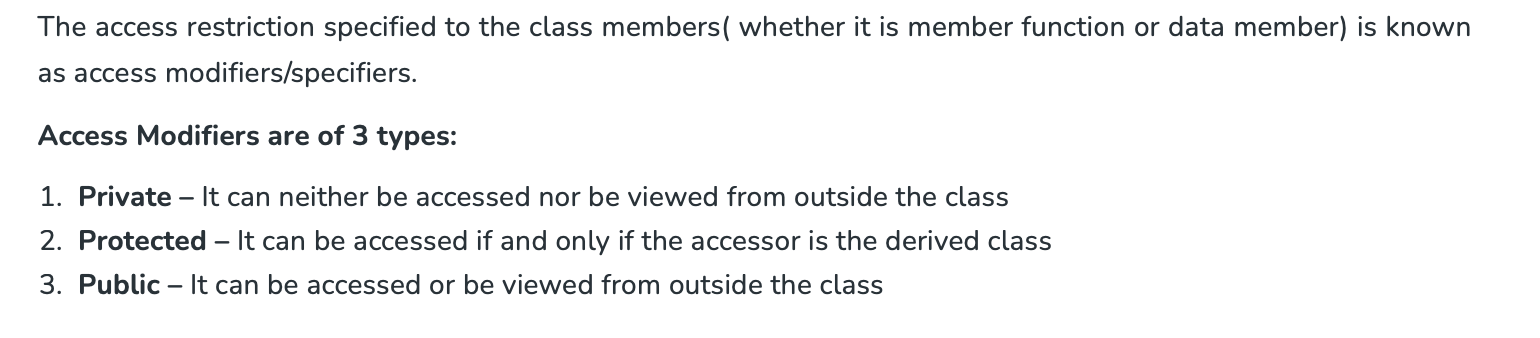
**Q12. What is the difference between a while loop and a do-while loop?**

**Ans.**

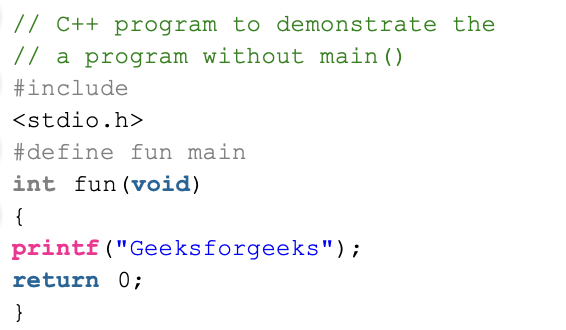
### **Q13. Discuss the difference between prefix and postfix?**

**Ans.**

**Q14.** **What are the C++ access modifiers?**

**Ans. **

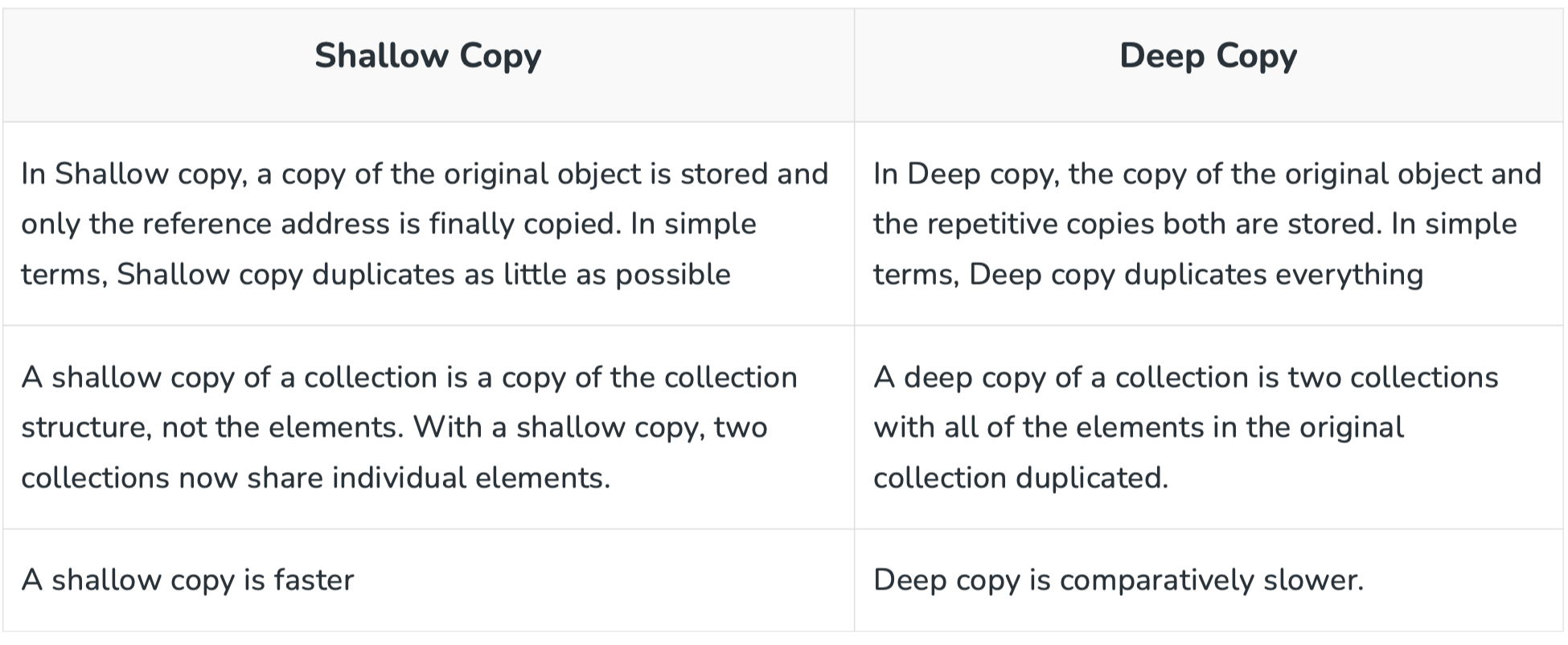
**Q15. Can you compile a program without the main function?**

**Ans.** Yes, it is absolutely possible to compile a program without a main(). For example Use Macros that defines the main.

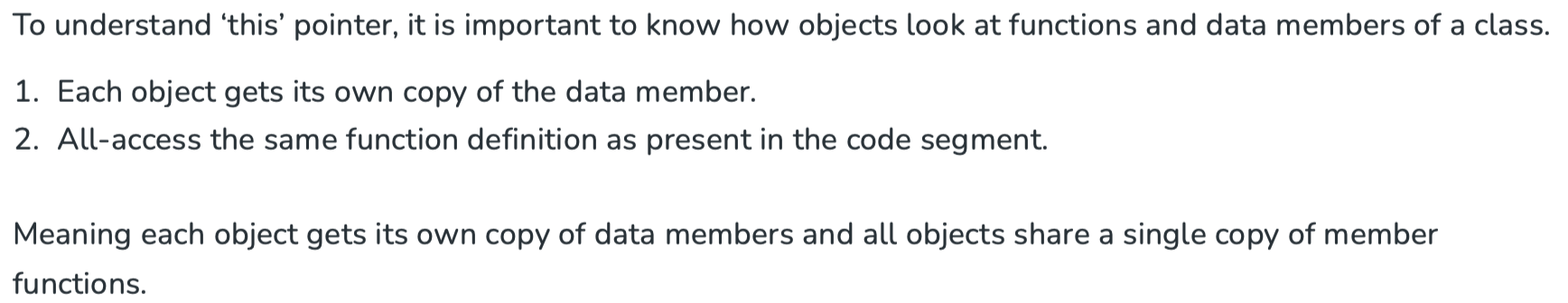
**Q16. What is the function of the keyword “Auto”?**

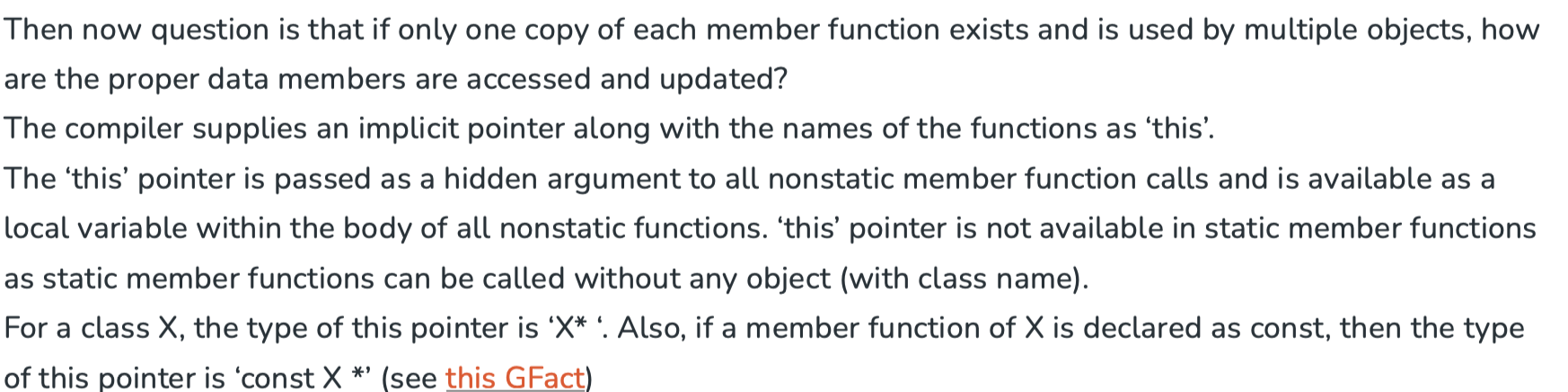
**Ans.** The **auto** keyword specifies that the type of the variable that is being declared will be automatically deducted from its initializer. In the case of functions, if their return type is auto then that will be evaluated by return type expression at runtime. Good use of auto is to avoid long initializations when creating iterators for containers. We can spend less time having to write out things the compiler already knows.

**Q17. What is Shallow Copy and Deep Copy?**

**Ans.**

**Q18. What is ‘*this*‘ pointer in C++?**

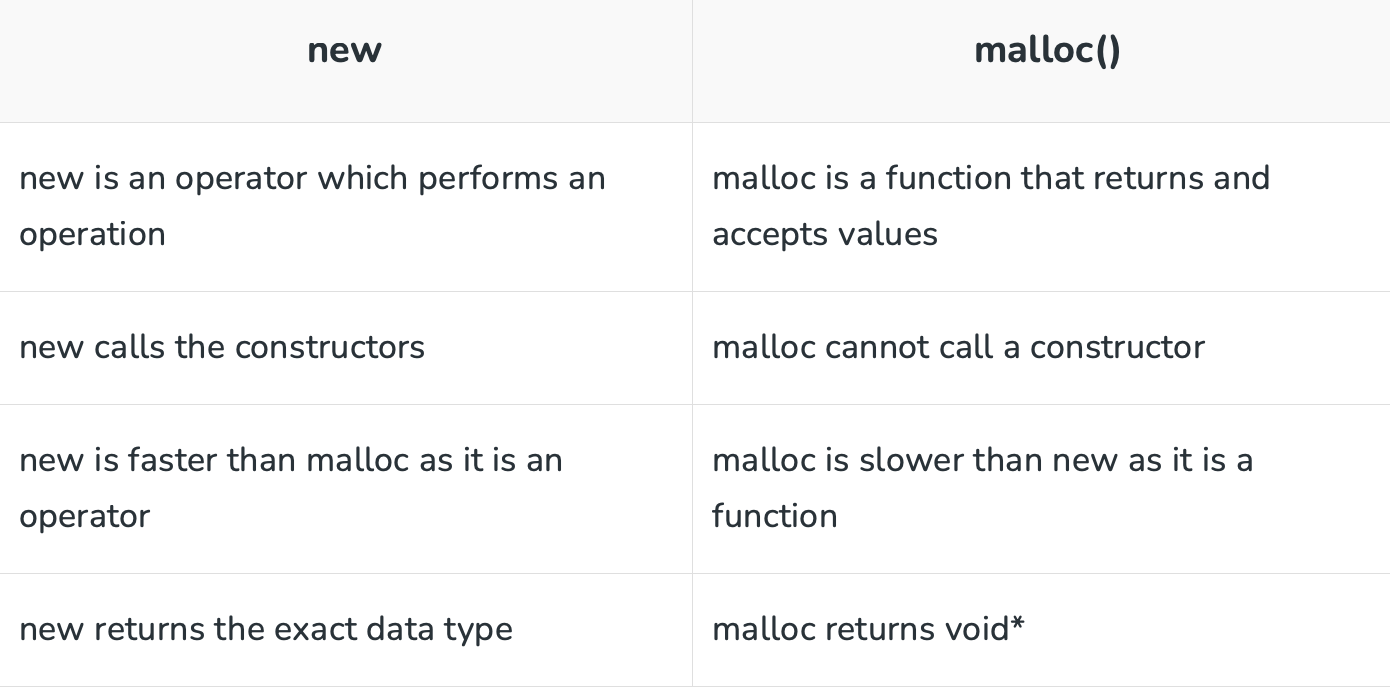
**Ans.**

****

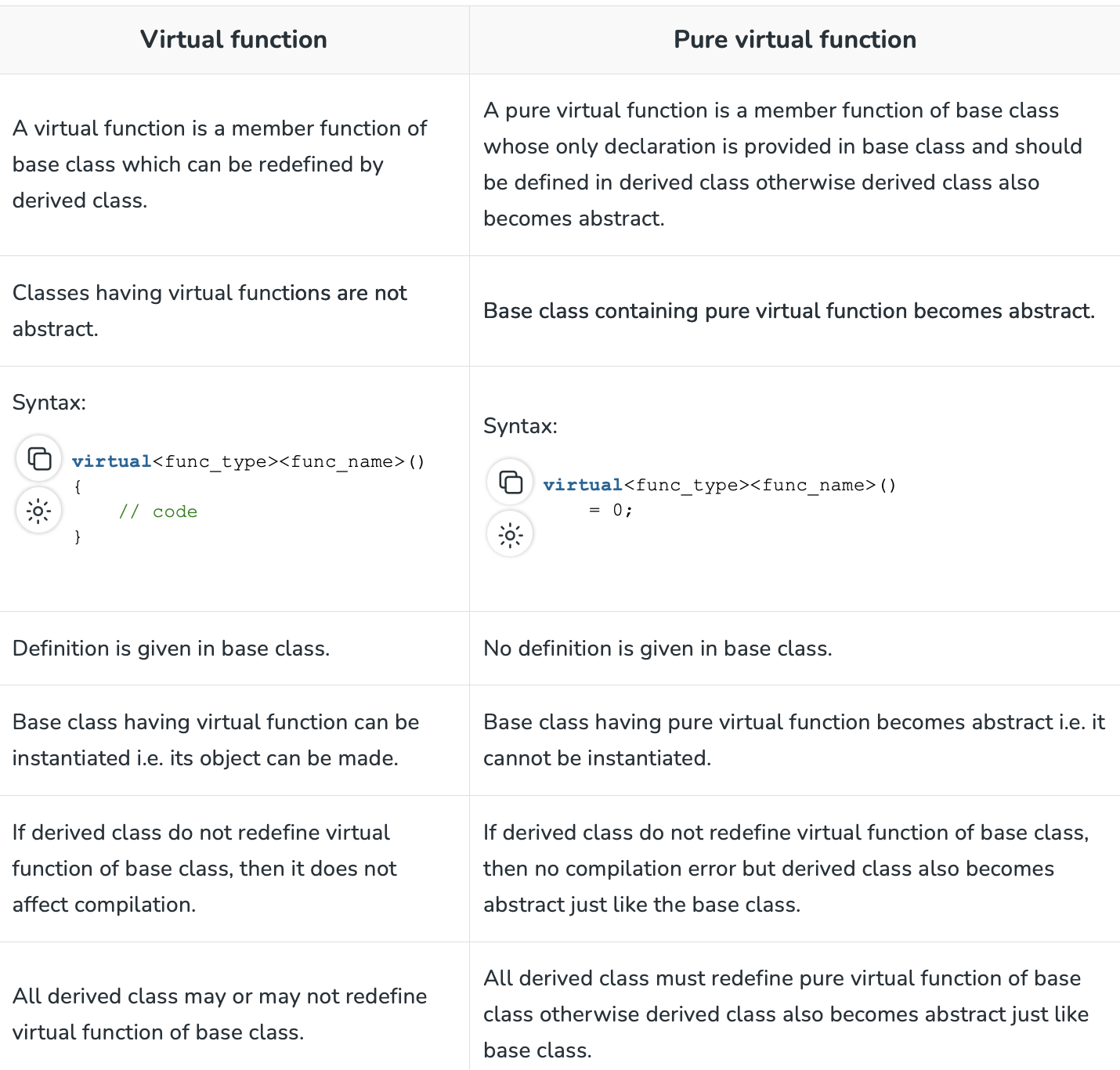
**Q19. What is function overriding ?**

**Ans.** When a function of the same name, same arguments or parameters, and same return type already present/declared in the base class is used in a derived class is known as Function Overriding. It is an example of Runtime Polymorphism or Late Binding which means the overridden function will be executed at the run time of the execution.

**Q20. What is the difference between new and malloc ()?**

**Ans.**

**Q21. What is the difference between virtual functions and pure virtual functions?**

**Ans.**

**Q22. What is Data Abstraction? What are abstract classes?**

**Ans.** Data Abstraction is the concept of OOP, it refers to providing only essential data to the outside world and hiding the background details. It is the concept of OOP, it refers to providing only essential data to the outside world and hiding the background details.

By definition, an **abstract class in C++** is a class that has at least one pure virtual function (i.e., a function that has no definition). The classes inheriting the abstract class must provide a definition for the pure virtual function; otherwise, the subclass would become an abstract class itself.

Abstract classes are essential to providing an abstraction to the code to make it reusable and extendable. For example, a *Vehicle* parent class with *Truck* and *Motorbike* inheriting from it is an abstraction that easily allows more vehicles to be added.

**Q23. What are inline functions?**

**Ans.** Normally when a function is called the control is transferred where the code is defined but in case of inline function the compiler while compilation the whole code is copied where the function is called.

An inline function is defined same as function just inline keyword before function name.

Inline function is those which are expanded at each call during the execution of the program to reduce the cost of jumping during execution.

Inline function in C++ is an enhancement feature that improves the execution time and speed of the program.

**Q24. What are classes and objects in C++**

**Ans.** A class is a user-defined data type where all the member functions and data members are tailor-made according to demands and requirements in addition to which these all can be accessed with the help of an object. To declare a user-defined data type we use a keyword class.

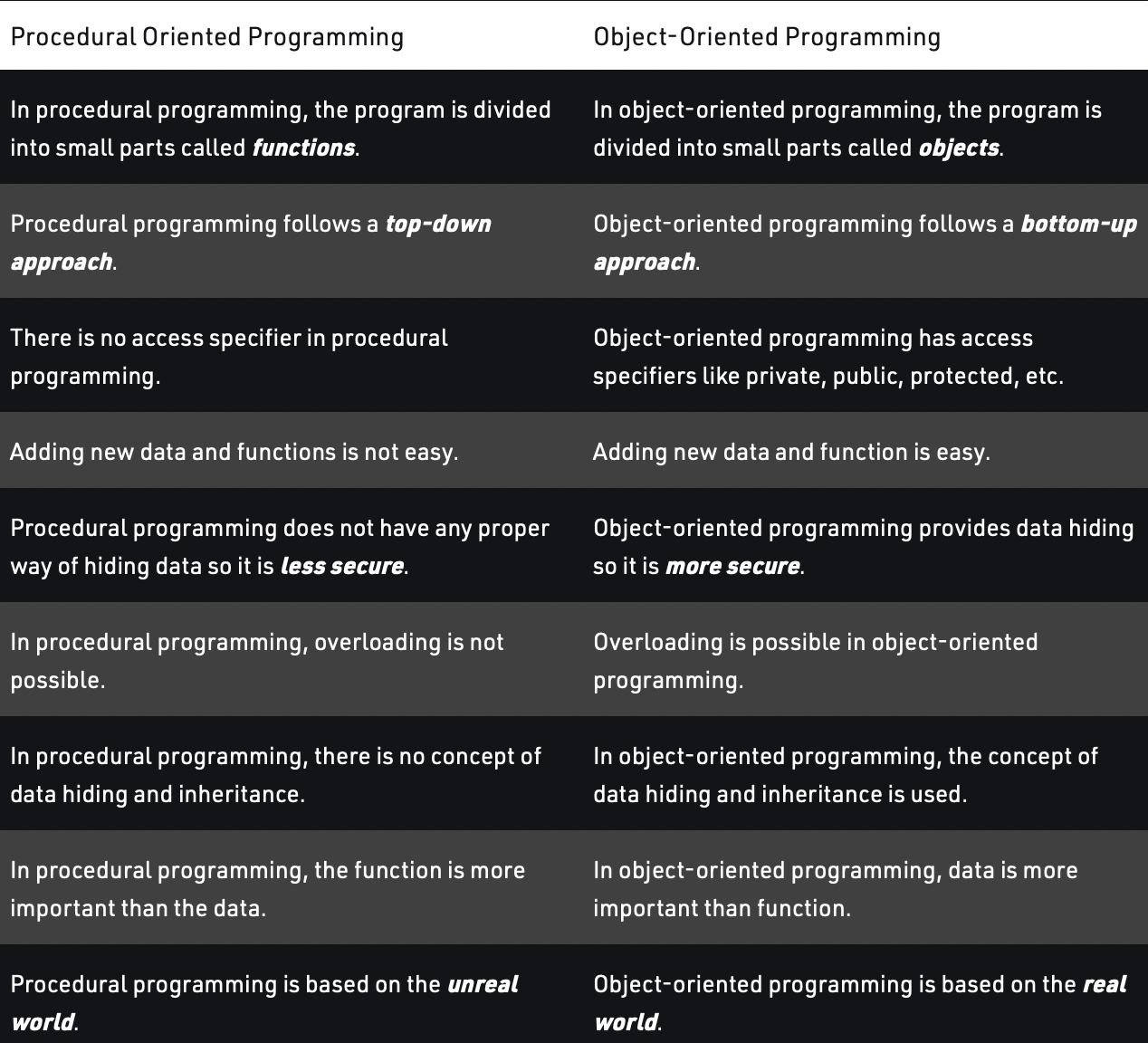
An object is an instance of a class and an entity with value and state; In simple terms, it is used as a catalyst or to represent a class member. It may contain different parameters or none.

**Q25. What are Abstract Data Types?**

**Ans.** Abstract Data Types (ADT) 🡪 ( Meaning of abstract is existing in thought or idea but lacking concrete existence ) these are the data types that we don’t need to know the inner structure. E.g. In case of stack we use pop() and push() operations but we are totally unaware of the inner implementation. @2 a class we create with different variables and operations like insert delete display all combined represents a ADT.

\*\* A friend function can access private members of a class

**Q26. What is procedural and object-oriented programming?**

**Ans.**

\*\*LIMITATIONS OF PROCEDURAL PROGRAMMING

🡪 If the structure of the data is changed then all the functions need to be changed to work in accordance to it.

**Q27. What is Polymorphism?**

**Ans.** Polymorphism is the ability to display a member function in multiple forms depending on the type of object that calls them.

In other words, we can also say that a man can be an employee to someone, a son of someone, a father of someone, and a husband of someone; this is how polymorphism can be projected in real life.

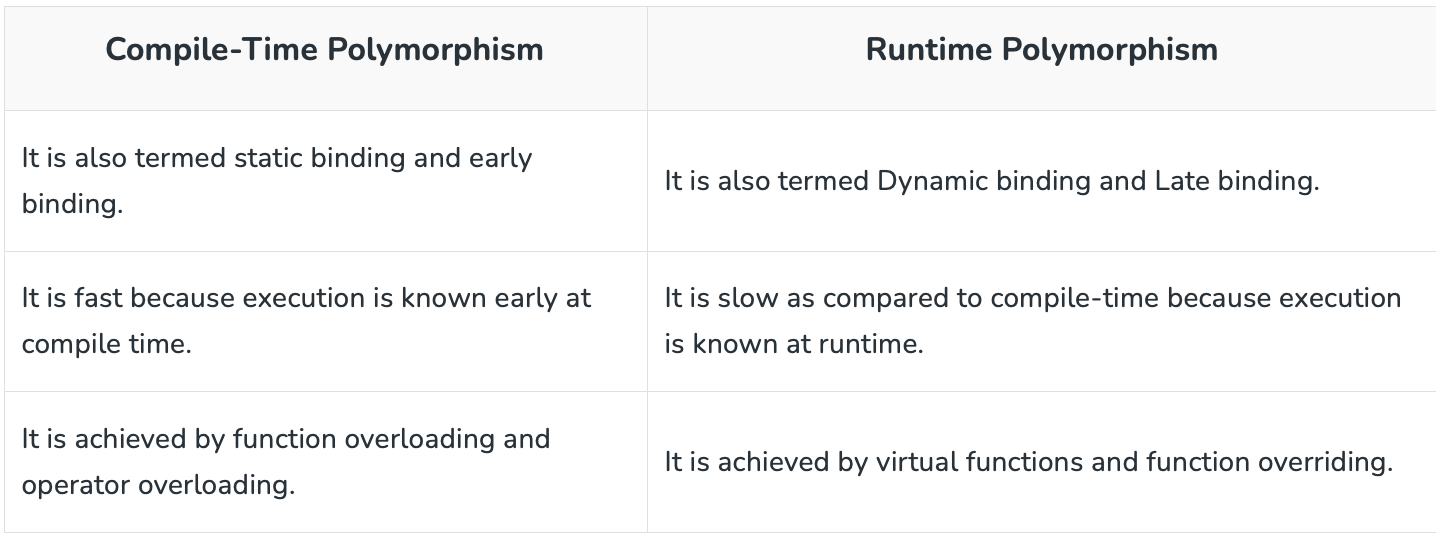
**Q28. What are the types of Polymorphism?**

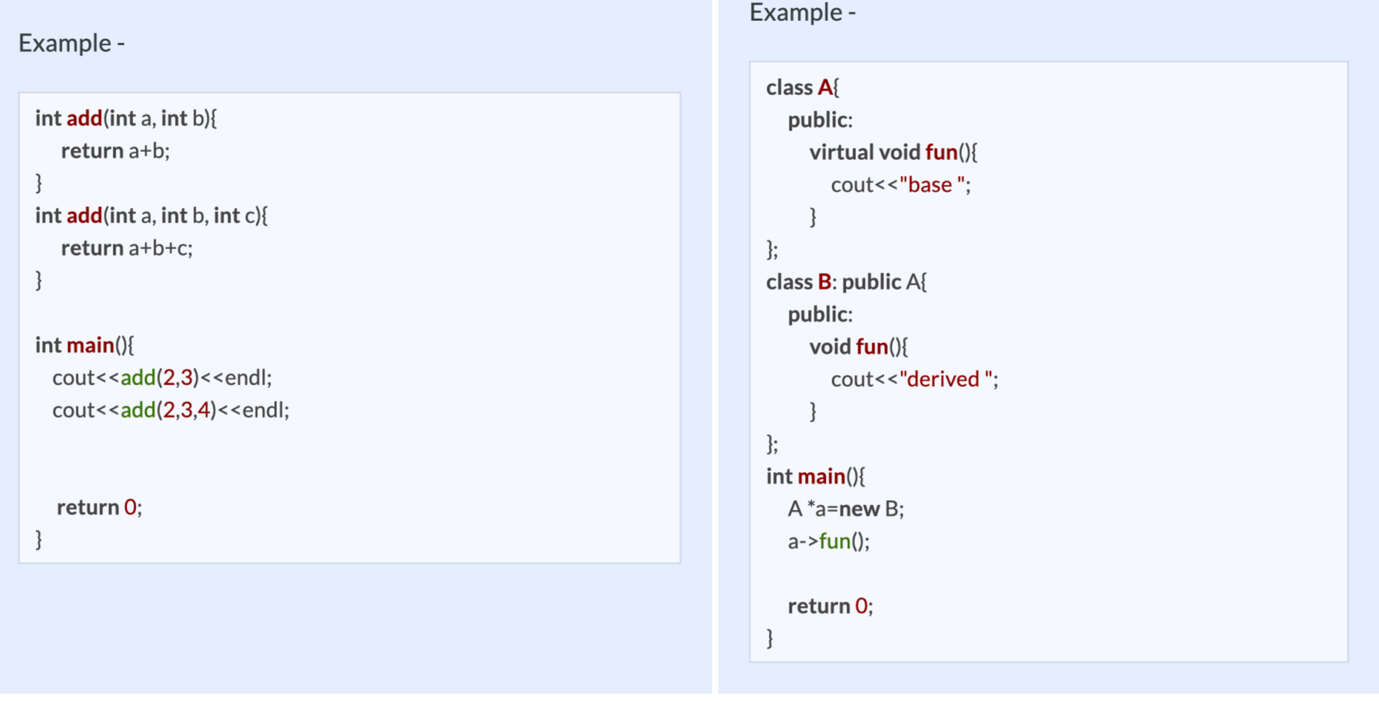
**Ans.** There are 2 types of polymorphism:

1. **Compile Time Polymorphism or Static Binding**

This type of polymorphism is achieved during the compile time of the program which results in it making a bit faster than Run time. Also, Inheritance is not involved in it. It is comprised of ***2 further techniques***: (a) Operator Overloading (b) Function Overloading

1. **Run-Time Polymorphism or Late Binding**

****This type of polymorphism is achieved by **Function Overriding**. Late binding and dynamic polymorphism are other names for runtime polymorphism.The function call is resolved at runtime in runtime polymorphism. In contrast, with compile time polymorphism, the compiler determines which function call to bind to the object after deducing it at runtime. (a) Virtual Functions

****

**Q29. What is Encapsulation?**

**Ans.** It is wrapping up data in a single unit. It is combination of data hiding and abstraction. It says the data members (variables) should be kept private and member functions should be used to manipulate data by this way the access of data member is limited to those who have access of the member functions

**Q30. What is Inheritance?**

**Ans.** Inheritance is the process of creating new classes, called derived classes, from existing classes. These existing classes are called base classes. The derived classes inherit all the capabilities of the base class but can add new features and refinements of their own.

**Virtual Inheritance**

Virtual inheritance is a technique that ensures only one copy of a base class’s member variables is inherited by grandchild-derived classes. Or in simple terms, virtual inheritance is used when we are dealing with a situation of multiple inheritances but want to prevent multiple instances of the same class from appearing in the inheritance hierarchy.

**Q31. What are constructors?**

**Ans.** A constructor is a special type of member function of a class, whose name is the same as that of the class by whom it is invoked and initializes value to the object of a class.

There are 3 types of constructors:

1. **Default constructor:**

It is the most basic type of constructor which accepts no arguments or parameters. Even if it is not called the compiler calls it automatically when an object is created.

1. **Parameterized constructor:**

It is a type of constructor which accepts arguments or parameters. It has to be called explicitly by passing values in the arguments as these arguments help initialize an object when it is created. It also has the same name as that of the class.

1. **Copy Constructor**

A copy constructor is a member function that initializes an object using another object of the same class. Also, the Copy constructor takes a reference to an object of the same class as an argument.

**Q32. What are Destructors?**

**Ans.** Destructors are members of functions in a class that delete an object when an object of the class goes out of scope. Destructors have the same name as the class preceded by a tilde (~) sign. Also, destructors follow a **down-to-top**approach, unlike constructors which follow a top-to-down.

~constructor\_name(); // tilde sign signifies that it is a destructor

**Virtual Destructors:**

When destroying instances or objects of a derived class using a base class pointer object, a virtual destructor is invoked to free up memory space allocated by the derived class object or instance.

Virtual destructor guarantees that first the derived class’ destructor is called. Then the base class’s destructor is called to release the space occupied by both destructors in the inheritance class which saves us from the memory leak. It is advised to make your destructor virtual whenever your class is polymorphic.

**Q34. What is Abstraction?**

**Ans.** Abstraction means only show relevant data and details rest of others are hide. This is the most important pillar in OOP. This is mostly done by*interfaces rather than abstract class*.

Eg.1 we use functions inside header files without knowing their functionality like pow, sine etc.

Eg.2 we have different cars but ignition button will start the car we don’t need to know the real implementation of the technology used inside.

**\*\*** Interface is a abstract class but it only contains purely virtual functions means all functions in it should be initialized to 0.

**Q35. Differentiate between Abstraction and Encapsulation.**

**Ans.**

**Q36. What are Friend Class and Friend Functions?**

**Ans.** A friend class is a class that can access both the protected and private variables of the classes where it is declared as a friend.

class Class\_1st {

// ClassB is a friend class of ClassA

friend class Class\_2nd;

statements;

}

class Class\_2nd {

statements;

}

A friend function is a function used to access the private, protected, and public data members or member functions of other classes. It is declared with a friend keyword. The advantage of a friend function is that it is not bound to the scope of the class and once it is declared in a class, furthermore to that, it cannot be called by an object of the class; therefore it can be called by other functions. Considering all the mentioned points we can say that a friend function is a global function**.**

**class GFG {**

**statements;**

**friend dataype function\_Name(arguments);**

**statements;**

**}**

**OR**

**class GFG{**

**statements'**

**friend int divide(10,5);**

**statements;**

**}**

**Q37. What does the Scope Resolution operator do?**

**Ans.** A scope resolution operator is denoted by a ‘::‘ symbol. Just like its name this operator resolves the barrier of scope in a program. A scope resolution operator is used to reference a member function or a global variable out of their scope furthermore to which it can also access the concealed variable or function in a program.

Scope Resolution is used for numerous amounts of tasks:

1. To access a global variable when there is a local variable with the same name
2. To define the function outside the class
3. In case of multiple inheritances
4. For namespace

**Q38. What is static int?**

**Ans.**

**Q39. What is Memoization?**

**Ans.** In programming, memoization is an optimization technique that makes applications more efficient and hence faster. It does this by storing computation results in cache, and retrieving that same information from the cache the next time it's needed instead of computing it again.

Eg. In case of recursive code of Fibonacci series we compute same values multiple times. With memoization, there's no need to recalculate the same values once and again – we just store each computation and return the same value when required again.

# **Q40.**

# **memset**(str, 't',3);

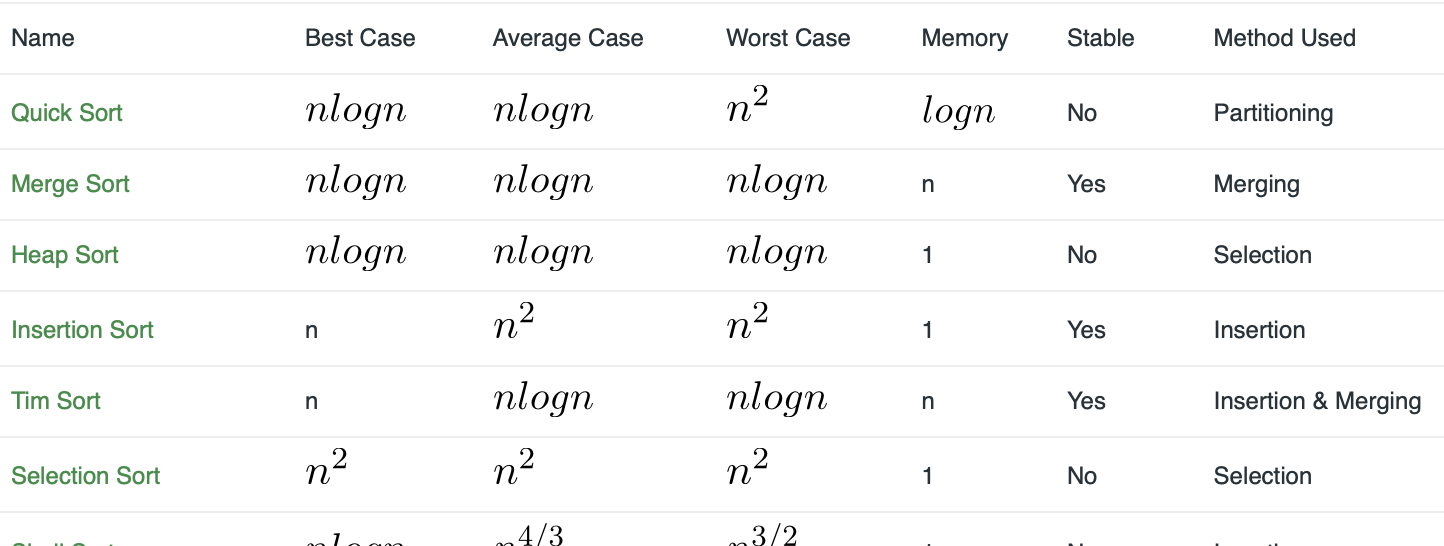
str=ttt

This takes varible as first argument and the value to be put inside In second and number of times the second argument is to be put in variable is put in 3rd argument.

# **Q41. Difference between subarray, subset & subsequence.**

**Ans.** For a given array = [1,2,3,4]

* Subarray/Substring : [1,2],[1,2,3] — is continuous and maintains relative order of elements
* Subsequence: [1,2,4] — is not continuous but maintains relative order of elements
* Subset: [1,3,2] — is not continuous and does not maintain the relative order of elements

\*\*Time complexity of algorithms where it gets divided by 2 its log 2 n

**Q41. What makes a sorting algorithm stable? What are factors that we consider while choosing a sorting technique?**

**Ans🡪** Stability of a sorting algorithm means in case of equal elements the order of occurrence is maintained.

Following factors are taken into consideration while opting for a sorting technique 🡪

1. Size of data
2. randomness of data
3. time and space complexity
4. stability

**Q42. What causes this instability?**

**Ans🡪** Swap of non-adjacent elements causes instability

**Q43. Why is significance order if the elements are the same?**

**Ans🡪** In the case of the leaderboard if two scores are the same then while sorting the first occurrence of the equal score must be kept above.

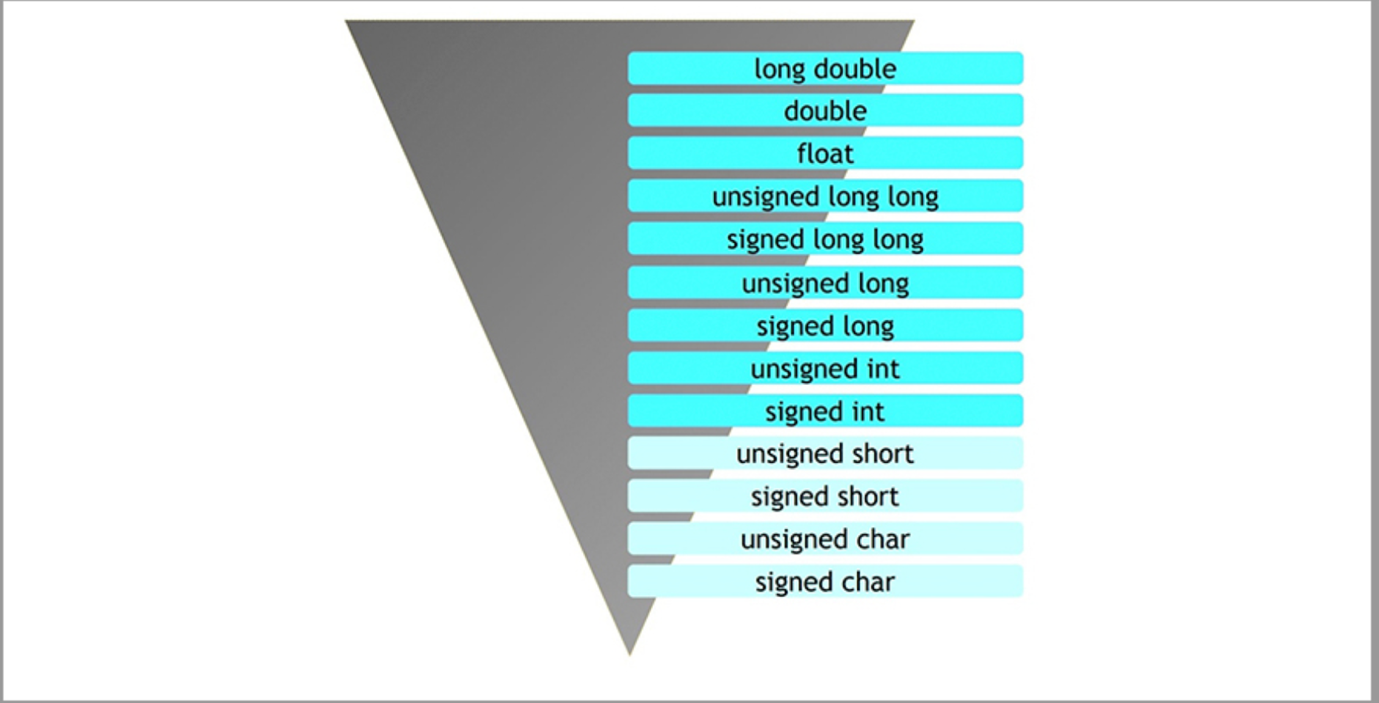
Q44. **What is Array Decay?**   
Ans. The loss of type and dimensions of an array is known as the decay of an array. This generally occurs when we pass the array into function by value or pointer. What it does is, it sends first address to the array which is a pointer, hence the size of the array is not the original one, but the one occupied by the pointer in the memory.

To prevent this decay pass size of array also as a parameter and not use sizeof on array parameters. Another way to prevent array decay is to send the array into functions by reference. This prevents the conversion of array into a pointer, hence preventing the decay.

\*\* While addition and other functions are cheaper in unordered map(O 1) than maps(O log n) , the best case time complexity and the average time complexity of deletion operation in an unordered\_map is O(1). While in the worst case, the time complexity of deletion operation in an unordered\_map is O(n). while in case of maps its O log n.

**Q45. What are ranks in data types?**

**Ans.** Whenever there are two operands of different data types in an expression, the operand with a lower rank will be converted to the datatype of the higher rank operand.

****

**\*\*\_\_gcd(6, 20)**

**Q46. What happens in print statement increment decrement statements?**

**Ans.** Make two list one with values and one with output. We know that i++ first prints and then increments so it is printed first even prior to the i. So increment for ++I and i++ store the value at every increment. Print occurs when i++ is found or all terms are travelled. Then print according to value.

**Q47. Maps and Red Black trees.**

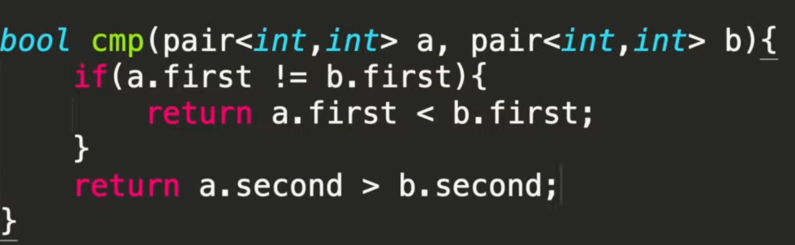
**Ans.** Maps STL uses Red Black trees which is a self-balancing tree. The elements in a map is in sorted manner.

In maps to take an element or insert an element the complexity is log N. Hence to print a N sized map the printing loop takes N log N.

**Q48. How does comparator function of the sort work?**

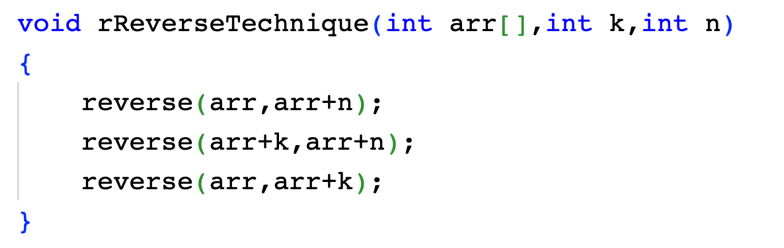
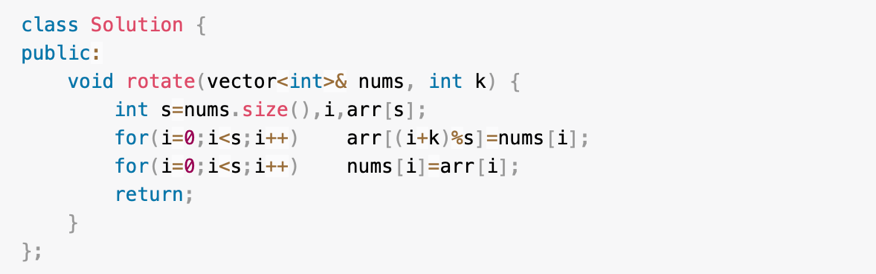
**Ans.** The comparator function of the sort swaps the element if false is returned else if true is returned it does nothing. Hence to make it work simply return the desired order of comparison . This works because it will not work if the given order is already there else it swaps.

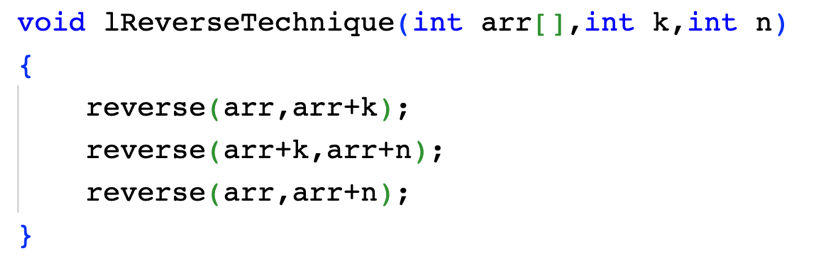
Eg.

****

**ARRAYS**

1. **How to cyclic rotate an array ‘k’ times?**
2. **Right**



1. **Left**
2. **What is the maximum size of the array that can be declared?**

**Ans:** If declared locally it can be maximum of 10^6 else if declared globally it can hold up to 10^7 elements.

if (not height.size())

1. **How to address rows and columns in 2D array?**

**Ans.** Let us consider a vector<vector<int>> mat. Number of rows is mat.size(), number of columns is mat[0].size(). m is row and n is column.

1. How to place any number to a position where it should be if the array is sorted?

Ans.

nth\_element(first\_iterator, iterator\_of \_target\_element, last\_iterator\_of\_container);

nth\_element(v, v + 4, v + 8); //5th element will be in sorted position.

Time Complexity of std::nth\_element(): O(n), with n being the distance between first and the last.

STRING

1. Find separate words in a string ?

Ans. Using stringstream using header file #include <sstream>

// Used to split string around spaces.

    istringstream ss(str);

    string word; // for storing each word

    // Traverse through all words

    // while loop till we get

    // strings to store in string word

**while** (ss >> word)

    {

        // print the read word

        cout << word << "\n";

    }

1. To convert anything into a string.

Ans. string str3 = to\_string(var1);

GENERAL

Q. What is RAID memory?.

Ans.

[Redundant Array of Independent Disks (RAID)](https://www.ontrack.com/en-us/data-recovery/raid) is a term used to describe computer storage systems that spread or replicate data across multiple drives.

There are two main reasons for RAID storage to work in this way: it increases data reliability and improves I/O (input/output) performance. That said, RAID storage isn't a perfect technology. Data loss can still occur when using it.