LearnNova

MOST ASKED - 40

Interview Questions

Q 1. What is C++?

Ans: As an extension of the C language, C++ was developed by Bjarne Stroustrup as a general purpose cross-platform language which gives programmers a high level of control over system resources and memory.

Q 2. What are the different Data Types present in C++?

Ans: There are 4 data types in C++:

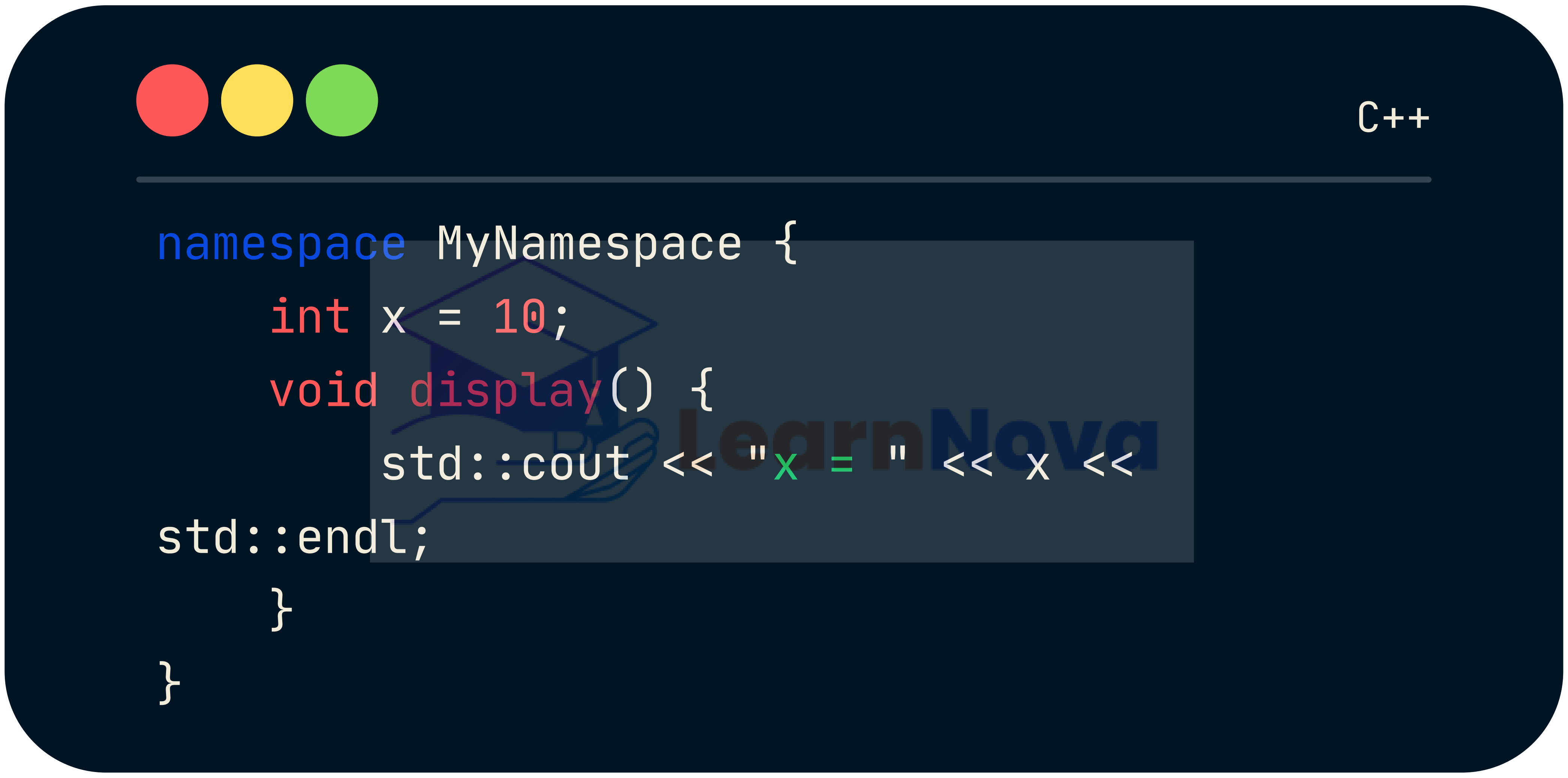
1. **Primitive Datatype** : e.g - char, short, int, float, long, double, bool, etc.
2. **Derived datatype** : e.g - array, pointer, etc.
3. **Enumeration** : e.g - enum
4. **User-defined data types** :e.g - structure, class, etc.

Q 3. What is operator overloading in C++?

Ans: An overloaded declaration is a declaration in the same scope of function or operator declared with the same name more than once.

Q 4. What is namespace in C++?

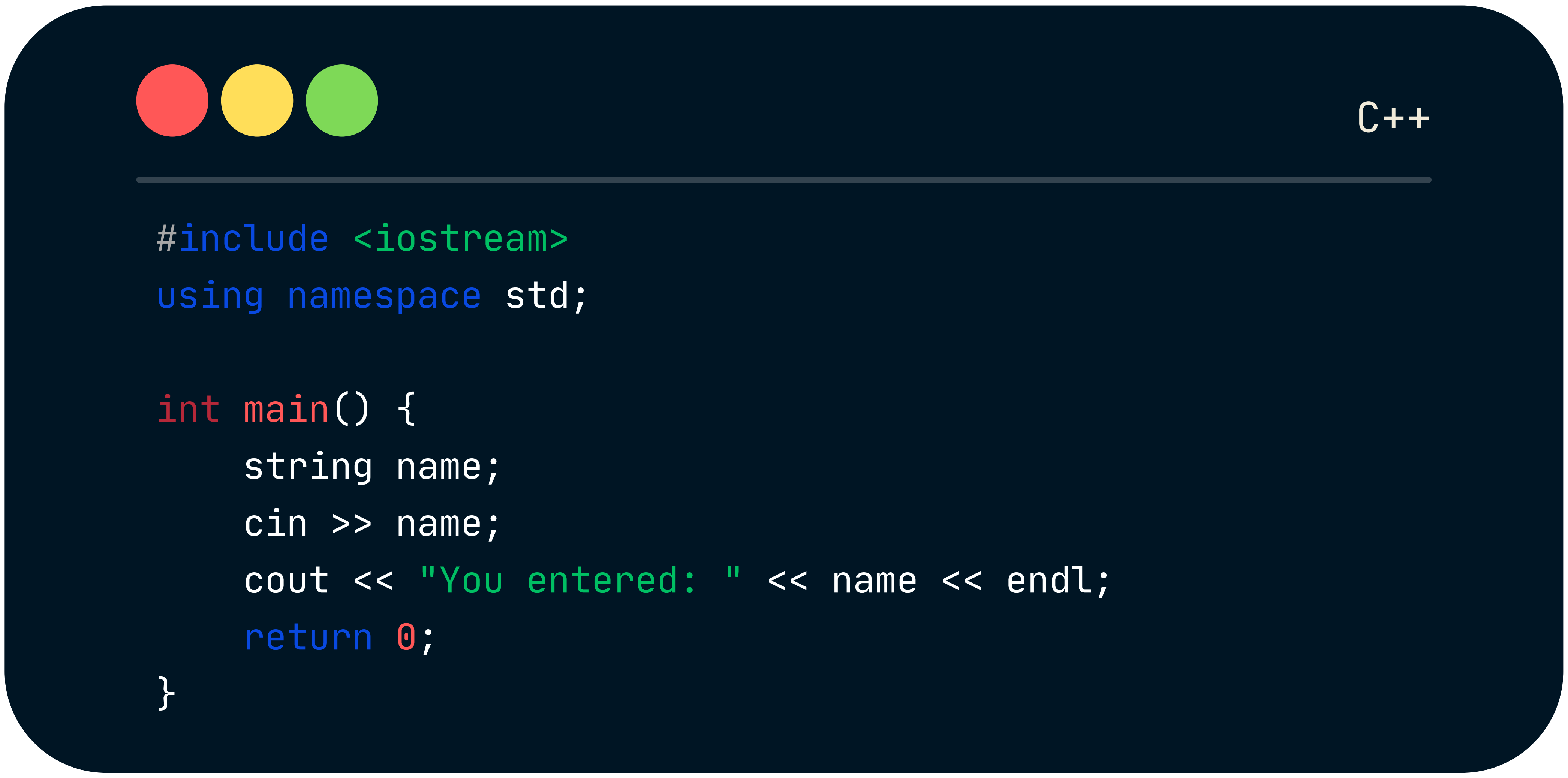
Ans: A namespace in C++ is a declarative region that provides scope to identifiers like variables, functions, and classes. It's mainly used to organize code and avoid name conflicts, especially when integrating multiple libraries.



```cpp
namespace MyNamespace {
    int x = 10;
    void display() {
        std::cout << "x = " << x <<
std::endl;
    }
}
```

Q 5. How to take input string in C++?

Ans:

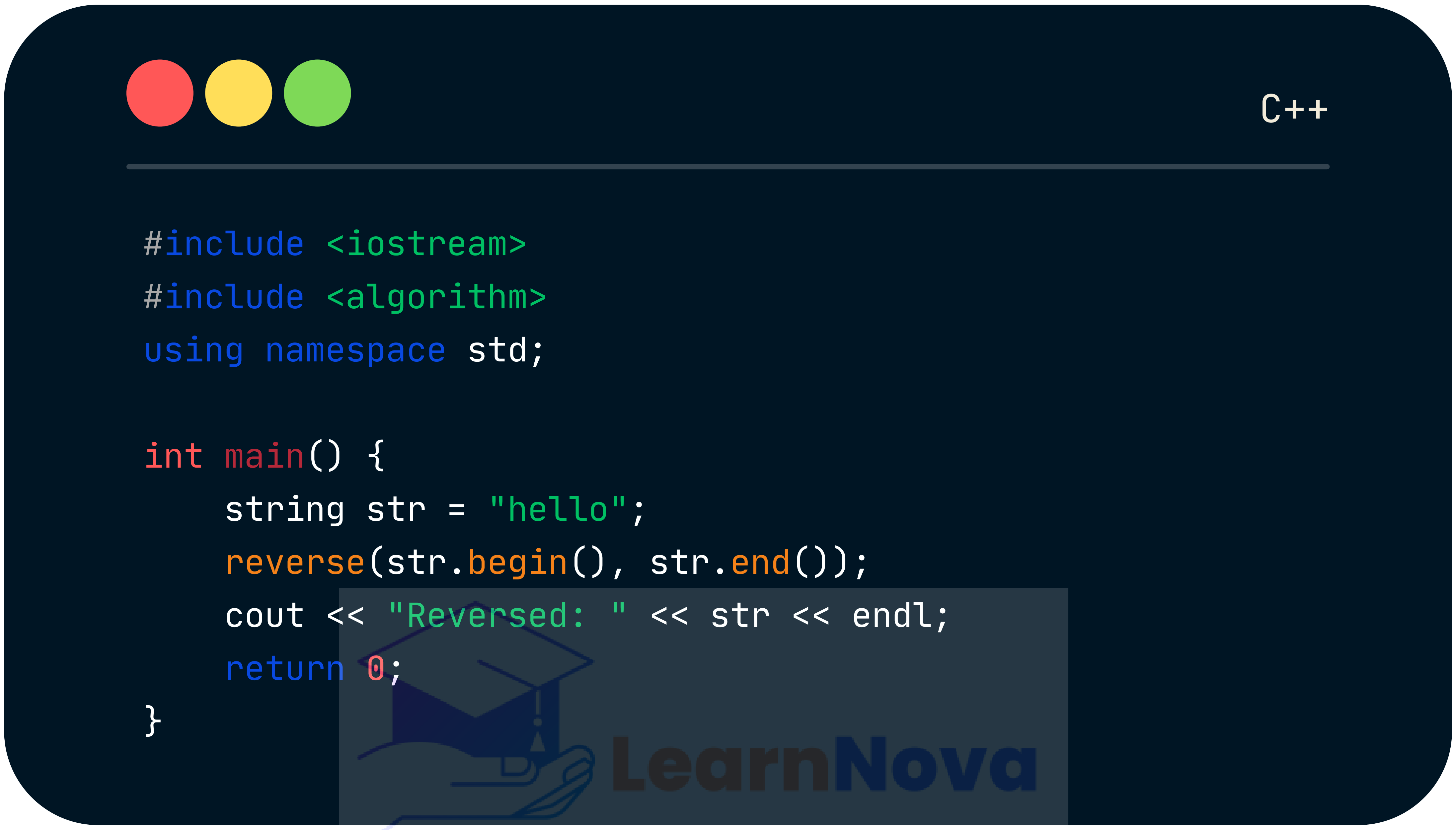


```cpp
#include <iostream>
using namespace std;

int main() {
    string name;
    cin >> name;
    cout << "You entered: " << name << endl;
    return 0;
}
```

Q 6. . How to reverse a string in C++?

Ans:

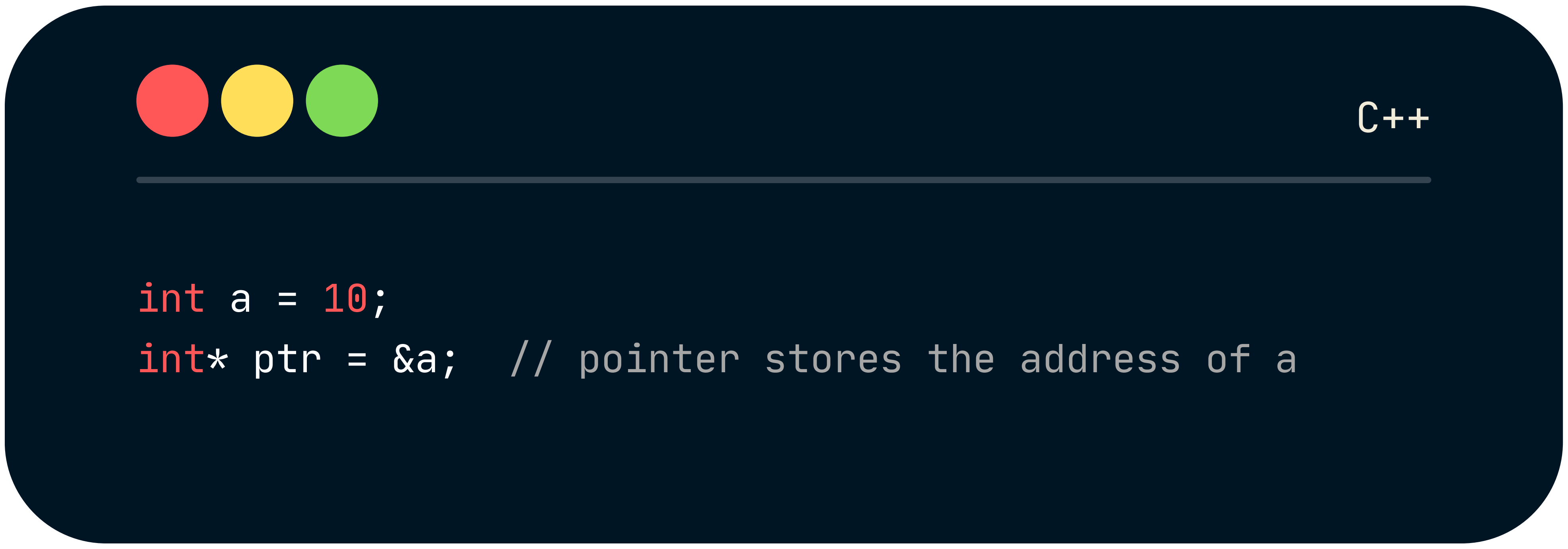


```cpp
#include <iostream>
#include <algorithm>
using namespace std;

int main() {
    string str = "hello";
    reverse(str.begin(), str.end());
    cout << "Reversed: " << str << endl;
    return 0;
}
```

Q 7. What is pointer in C++?

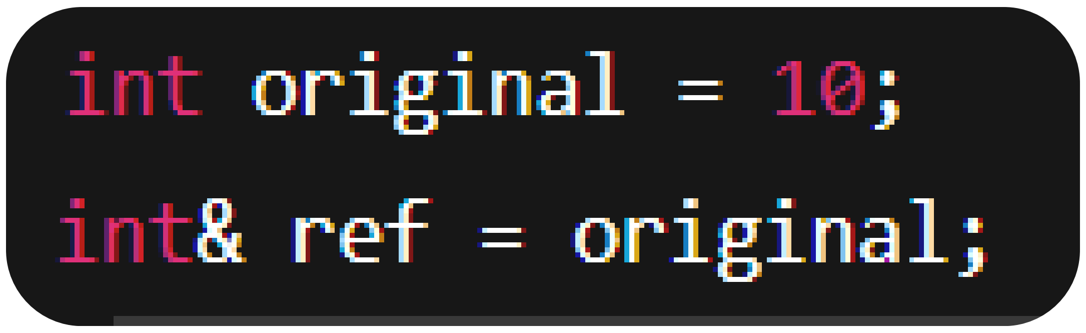
Ans: Pointers in C++ are a data type that store the memory address of another variable.



```cpp
int a = 10;
int* ptr = &a;   // pointer stores the address of a
```

Q 8. What is a reference variable in C++?

Ans: A reference variable acts as an alias for an existing variable. When a variable is declared as a reference, it becomes another name for the original variable, meaning any operations performed on the reference are actually performed on the original variable.



```
int original = 10;
int& ref = original;
```

Q 9. What is the difference between C and C++?

Ans:

C	C++
Procedural programming language	Supports both procedural and object-oriented programming
No support for classes and objects	Supports classes and objects (OOP features)
No function or operator overloading	Supports function and operator overloading
Uses printf() and scanf() for I/O	Uses cin and cout for I/O
Manual memory management (malloc, free)	Uses new and delete (also supports constructors/destructors)

Q 10. What is function in C++?

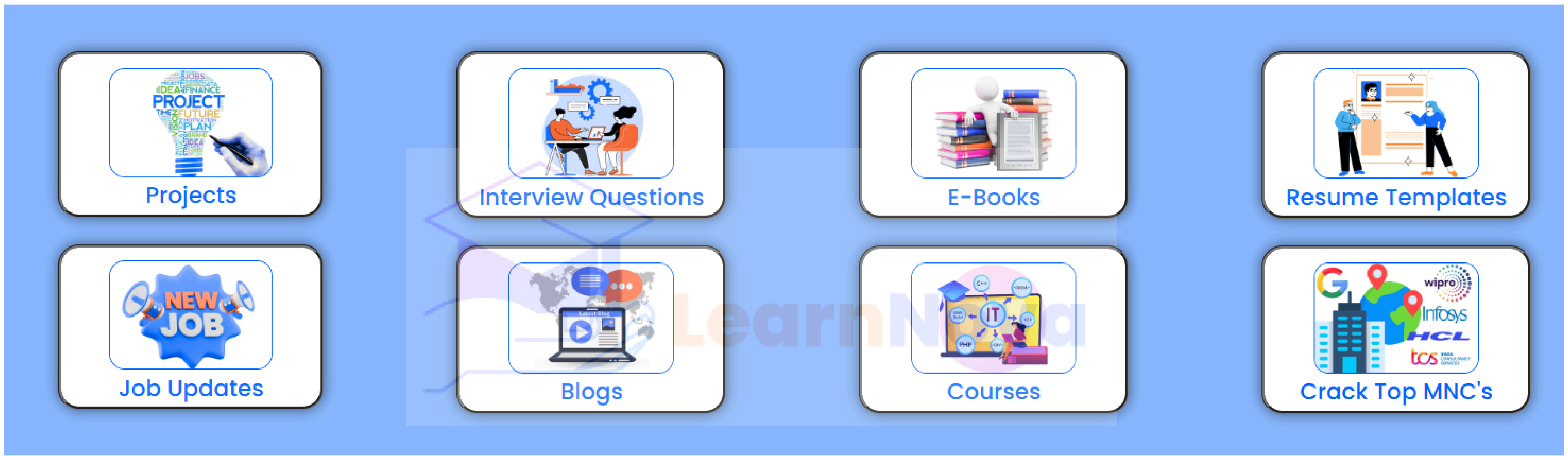
Ans: A function in C++ is a reusable block of code that performs a specific task. It has a return type, a name, optional parameters, and a body. Functions help in organizing code, improving readability, and avoiding repetition. For example, a function can take inputs, process them, and return a result. C++ supports various types of functions including user-defined, inline, recursive, and overloaded functions.

Q 11. What is destructor in C++

Ans: Destructors in c++ are special function/methods that are used to remove memory allocation for objects. They are called usually when the scope of an object ends. eg. when a function ends you can call a destructor. They are of the same name as the class – syntax – ~();

Q 12. What is function overloading in C++

Ans: Function Overloading happens in C++ when two or more functions share the same name. They can be differentiated on the basis of the type of data they are passing as parameters or even the number of paramters they are passing. eg. int fun(char a); & int fun(int b); & void fun(int a, int b)



“

You don’t have to see the whole staircase, just take the first step.

Martin Luther King, Jr.

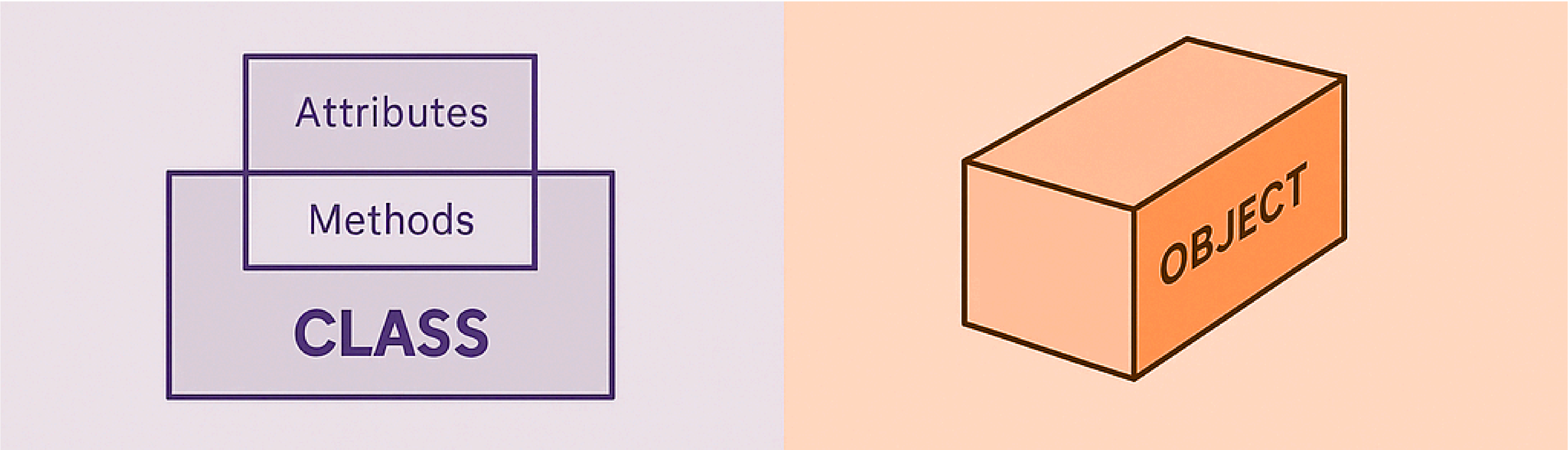
Q 13. What are Classes and Objects in C++?

Ans: **Class**:

- Blueprint: A class is a blueprint or a template for creating objects.
- Data and Functions:The encapsulation of data (attributes) and functions (methods) that operate on the data.
- Abstraction: Provides a way to represent real-world entities with properties, behaviors, and expressions.

Object:

- Instance: An object is an instance of a class representing a specific entity.
- Attributes: Objects store data defined in the class's attributes.
- Methods: Objects can invoke the methods defined in the class, performing operations specific to that object's type.



Q 14. What is exception in C++

Ans: Runtime abnormal conditions that occur in the program are called exceptions.

These are of 2 types:

- Synchronous
- Asynchronous

C++ has 3 specific keywords for handling these exceptions:

- – try – catch – throw

Q 15. How to get absolute value in C++

Ans: There are three functions in the cstdlib header file to return the absolute value of the integer.

Those are: – abs() – labs() – llabs()

The difference lies in the range for integer value being passed as an argument. For abs() its type int in C++. For labs(), its type long int in C++ and for llabs() its long long int in C++.

Q 16. What is stack in C++?

Ans: A linear data structure which implements all the operations (push, pop) in LIFO (Last In First Out) order. Stack can be implemented using either arrays or linked list.

The operations in Stack are

- Push: adding element to stack
- Pop: removing element from stack
- isEmpty: returns true if stack is empty
- Top: returns the top most element in stack

Q 17. Define token in C++

Ans: A token is the smallest unit of code that the compiler recognizes during compilation.

- **Identifiers** – Names of variables, functions, classes, etc.
- **Keywords** – Reserved words like if, int, while.
- **Literals** – Constant values like numbers (42) or strings ("text").
- **Operators** – Symbols like +, -, *, &&.
- **Comments** – Notes in code (//, /* */) that are ignored by the compiler.
- **Whitespace** – Spaces, tabs, and newlines that separate tokens but aren't processed.

Q 18. What is stream in C++?

Ans: Stream refers to a stream of characters to be transferred between program thread and i/o.

Q 19. What is Operator Overloading in C++?

Ans: Operator overloading in C++ lets you redefine the behavior of operators (like +, -, *) for user-defined types.

It allows class objects to behave like built-in types by providing custom implementations for these operators.

Q 20. What is Polymorphism in C++?

Ans: Polymorphism in C++ allows objects of different classes to be treated as objects of a common base class. It enables functions to work with different derived class objects through base class pointers or references, making the code more flexible and reusable.

C++ supports two types of polymorphism:

- **Compile-time polymorphism** (e.g., function overloading, operator overloading)
- **Runtime polymorphism** (achieved using virtual functions)

Q 21. Explain the Constructor in C++

Ans: In C++, a constructor is a special function that has the same name as the class. It runs automatically when you create an object and is used to initialize the object's data members, making sure the object starts in a valid state.

Q 22. What is polymorphism?

Ans: **Compile-Time Polymorphism**

- Also known as: Static binding or early binding
- Achieved using: Function overloading and operator overloading
- Decision made at: Compile time
- Performance: Faster (no runtime overhead)
- Flexibility: Less flexible, behavior fixed at compile time

Runtime Polymorphism

- Also known as: Dynamic binding or late binding
- Achieved using: Inheritance and virtual functions
- Decision made at: Runtime
- Performance: Slightly slower due to virtual table lookup
- Flexibility: More flexible, supports dynamic behavior

Q 23. What is enum in C++?

Ans: **enum** is abbreviation of Enumeration which assigns names to integer constant to make a program easy to read.

Syntax for the same:
enum enum_name{const1, const2, ... };

Q 24. What do you mean by Abstraction in C++?

Ans: Abstraction in C++ means hiding the internal implementation details and showing only the essential features of an object. It helps simplify code and allows you to focus on what an object does, not how it does it.

Q 25. What are Destructors in C++?

Ans: A destructor is a special function in a class that cleans up when an object is no longer needed. It has the same name as the class but starts with a tilde (~). The destructor is automatically called when an object goes out of scope or is deleted, helping to free resources like memory or file handles to prevent leaks.

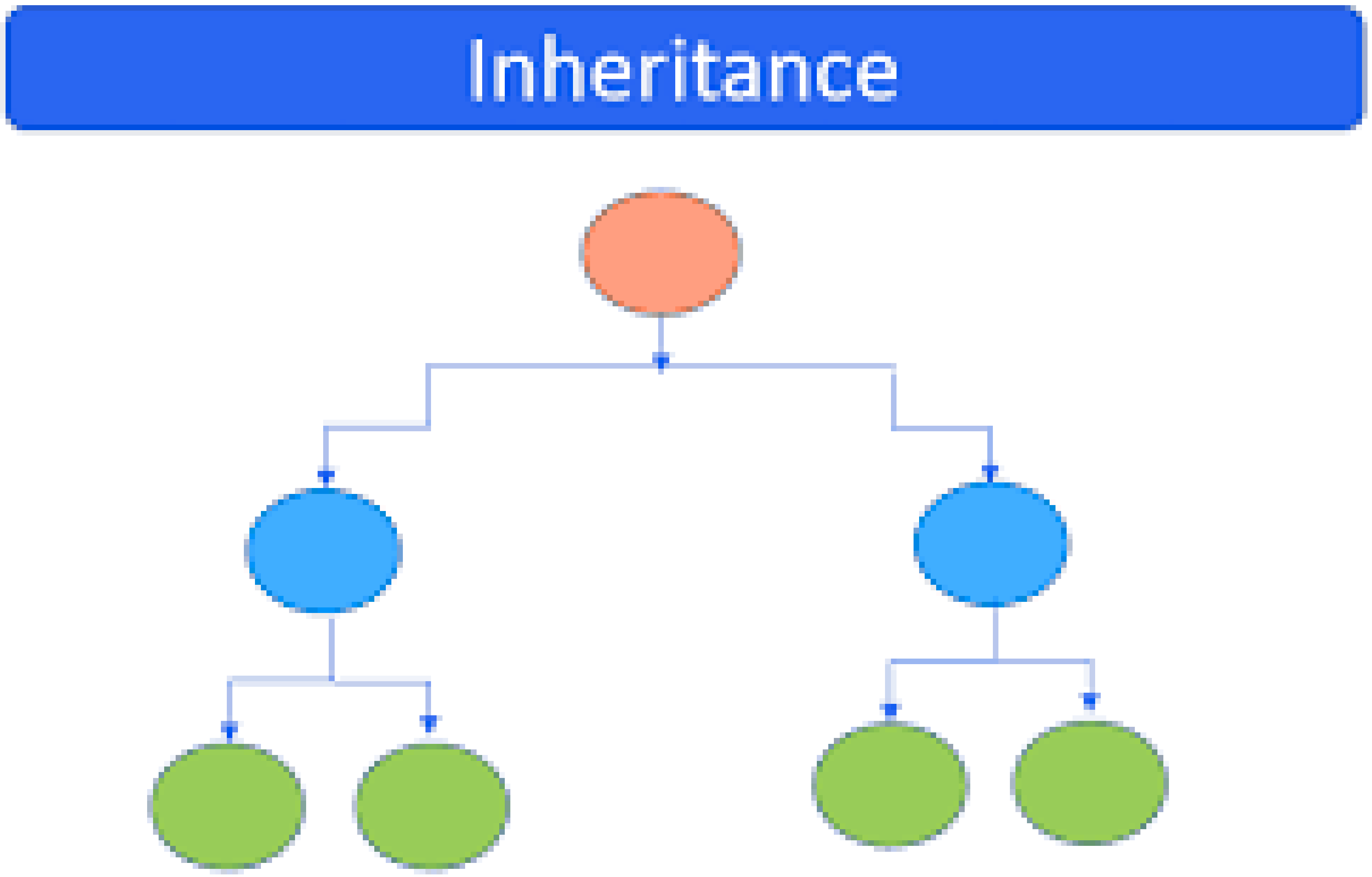
Q 26. Can Java support multiple inheritance?

Ans: Java does not support multiple inheritance with classes to avoid ambiguity.

it allows multiple inheritance using interfaces.

Q 27. Explain Inheritance in C++

Ans: Inheritance in C++ is a way for one class to take on the properties and behaviors of another class. The child class inherits from the parent class, which means it can use the parent's functions and variables without having to rewrite them. This helps save time, avoid code duplication, and makes it easier to organize related classes in a clear hierarchy.



Q 28. What is Function Overriding in C++?

Ans: Function overriding in C++ occurs when a derived class defines a function with the same name, return type, and parameters as one in its base class. This lets the derived class provide its own behavior, which is called instead of the base class version, enabling polymorphism.

Q 29. What is the purpose of the “delete” operator in C++?

Ans: The ***delete*** operator in C++ frees memory allocated by ***new*** and calls the destructor to clean up the object.

Q 30. What are character constants in C++

Ans: Character constant are members of the character set in which a program is written which is surrounded by single quotation marks (‘).

Q 31. What is flush in C++?

Ans: std::flush synchronizes the stream buffer with its controlled output sequence

Q 32. What is the difference between new and malloc() in C++?

Ans:

- **new** is an operator that allocates memory and also calls the constructor to initialize objects. It returns a pointer of the appropriate type and throws an exception if allocation fails.
- **malloc()** is a C library function that only allocates raw memory but does not call constructors. It returns a void* that needs to be cast to the appropriate type and returns nullptr (or NULL) if allocation fails.

Q 33. Can you compile a program without the main function?

Ans: A program may be compiled without a main() call. For instance, Use Macros that define the main

Q 34. What is the difference between == and = in C++?

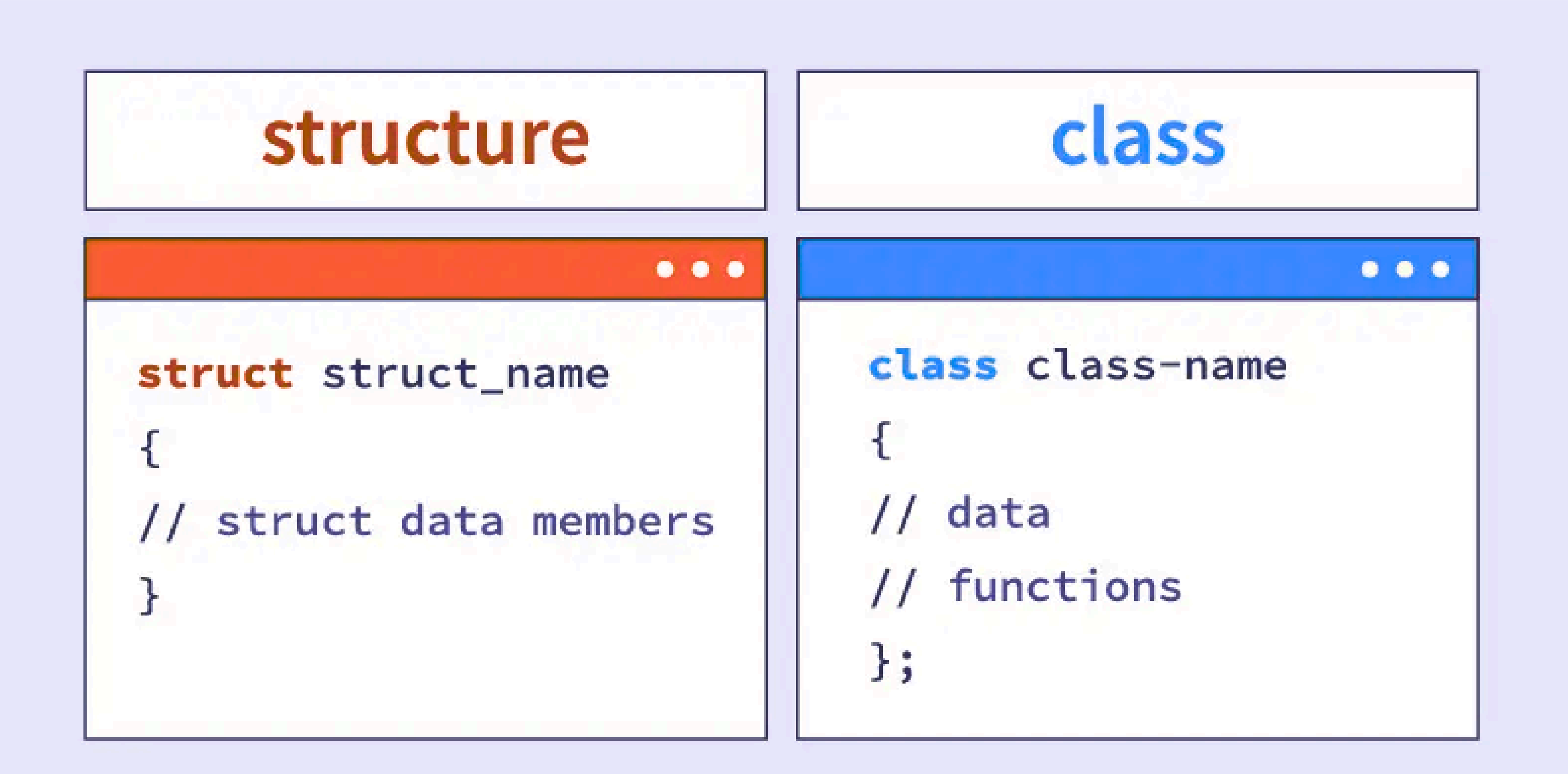
Ans: **=** is the assignment operator (used to assign values), while **==** is the equality operator (used to compare values).

Q 35. What is the purpose of a virtual function in C++?

Ans: A virtual function allows derived classes to override it so that the correct function is called through a base class pointer or reference, enabling runtime polymorphism.

Q 36. What is the difference between struct and class in C++?

Ans: By default, members of a struct are public, meaning they can be accessed from outside the struct. In contrast, members of a class are private by default, restricting access only within the class itself. Apart from this default access difference, struct and class are essentially the same in C++.



```
struct struct_name
{
// struct data members
}
```

```
class class-name
{
// data
// functions
};
```

Q 37. What are the four main principles of Object-Oriented Programming (OOP) in C++?

Ans: The four main OOP principles in C++ are:

1. **Encapsulation** – Bundling data and functions that operate on that data within a class, restricting direct access to some of the object's components.
2. **Abstraction** – Hiding complex implementation details and exposing only the necessary features.
3. **Inheritance** – Creating new classes from existing ones to promote code reuse and establish relationships.
4. **Polymorphism** – Allowing functions or operators to behave differently based on the object's actual type, mainly through function overloading (compile-time) and virtual functions (runtime).

Q 38. What is the difference between ++i and i++?

Ans: **++i** (pre-increment) increments the value first and then returns it, while **i++** (post-increment) returns the value first and then increments it.

Q 39. What is a mutable storage class specifier? How can they be used?

Ans: The mutable keyword allows a class member to be modified even if the object is declared const. This is useful because normally const objects can’t change their members. However, mutable can’t be used with reference, static, or const members. In a const member function, only mutable members can be modified since the this pointer is treated as a pointer to a const object.

Q 40. Define storage class in C++ and name some of them

Ans: The properties (lifetime and visibility) of a variable or function are defined by the storage class. These properties often aid in tracking a variable's existence while a program runs.

Syntax :-

```
storage_class var_data_type var_name;
```

Our students have gone on to work at renowned companies, innovative startups, and leading unicorns.

tcs TATA CONSULTANCY SERVICES

Infosys

wipro

HCL

Capgemini

cognizant

make my trip

ZOHO

Tech Mahindra

amazon

facebook

Explore our Programs

LEARN NOVA
Project
Job Updates
Interview Questions
Blogs
E - Books
Courses
Resume Templates
Crack Top MNC's