

Mastering C++: From Basics to Advanced Projects



runner-code.com



Presented By Runner Code



Table of Contenes

1-Introduction to C++
2-Variables, Data Types, and Operators
3-Control Statements
4-Functions and Recursion
5-Arrays and Strings
6-Object-Oriented Programming (OOP)
7-Pointers and Dynamic Memory
8-File Handling
9-Templates and Standard Template Library (STL)
10-Advanced Projects



Defines the title in the browser bar
Derive a name for the page when bookmarked
Displays the name of the page when bookmarked





C++ is a powerful, versatile, and efficient programming language widely used in various applications, from system/software development to game development and scientific computing. It is an extension of the C language, designed by Bjarne Stroustrup in the early 1980s, to incorporate objectoriented programming features while retaining the speed and low-level capabilities of C.

Introduction

- Key Features of C++:
- 3. Portability:
- 4. Rich Standard Library:
- 5. Memory Management:
- 6. Scalability:

```
1.Object-Oriented Programming (OOP):

    Supports classes, objects, inheritance,

     polymorphism, encapsulation, and abstraction,
     making it ideal for complex software
     development.
2. High Performance:
   • Combines low-level programming capabilities with
     high-level abstraction, ensuring efficiency in
     resource-constrained environments.
   • Code written in C++ can run on multiple
     platforms with minimal modification.
   • Includes the Standard Template Library (STL) for
     data structures, algorithms, and iterators.
   • Provides fine-grained control over system
     resources through manual memory management with
     pointers and dynamic allocation.
   • Suitable for projects of all sizes, from small
     programs to large-scale enterprise applications.
7. Multi-Paradigm Support:
   • Supports procedural, object-oriented, and
     generic programming styles.
```

Ray Learn Chr 2





- System Development: Used to build operating systems, databases, and compilers.
- Game Development: The language of choice for developing high-performance games.
- Scientific Applications: Ideal for simulations and mathematical computations.
- Foundational Knowledge: Understanding C++ provides a strong base for learning other languages like C#, Java, or Python.

C++ remains one of the most popular and influential programming languages in the world, known for its combination of speed, flexibility, and precision. Whether you're developing a game, working on embedded systems, or delving into competitive programming, C++ equips you with the tools to succeed.

Setting Up the Development Envilonment

Visual Studio). access.









1. Download and install a compiler (e.g., GCC, 2.Use an IDE (e.g., Code::Blocks, Visual Studio Code) or a text editor with terminal

3. Write and compile your first program.



MOUTE FEEST CEE Program

Coce Example:

#include <iostream> using namespace std;

```
int main() {
    cout << "Welcome to C++ Programming!" << endl;</pre>
    return 0;
```

3

Explanation:

- completion.



• #include <iostream>: Includes the library for input/output operations.

• using namespace std;: Simplifies access to standard functions.

• cout: Prints text to the console.

• return 0;: Signals successful program

Chapter 2: Variables, Data Types, and Operators Variables and Data Types

Data Types: int, float, double, char, bool, string. • Example:

int age = 25;float height = 5.9; char grade = 'A'; bool isStudent = true;

Input and Output Operations Code Example:

#include <iostream> using namespace std; int main() { int age; cout << "Enter your age: ";</pre> cin >> age; cout << "You entered: " << age << endl;</pre> return 0;

Operators



- Arithmetic Operators: +, -, *, /, %
- Relational Operators: ==, !=, <, >, <=, >=
- Logical Operators: &&, ||, !



int a = 10, k
cout << (a +
cout << (a >
(false)







Chapter 3: Control Statements







if (age >= 18) {
 cout << "You are an adult." << endl;
} else {
 cout << "You are a minor." << endl;
}</pre>







Switch Case Example:

```
switch (grade) {
    case 'A':
        cout << "Excellent!";
        break;
    case 'B':
        cout << "Good!";
        break;
    default:
        cout << "Try harder!";
}</pre>
```











For Loop Example:

for (int i = 0; i < 5; i++) {</pre> cout << i << endl;</pre> }







BEGIN NAVIGATION <11> a href="index.html">Home Home Events Multiple Column Men <a href="#" class="current"; Tall But Image Logo class="active">Ta class="has-children"> Carousels



While Loop Example:

```
int i = 0;
while (i < 5) {
    cout << i << endl;
    i++;
}
```





Do-Whfile Loop Example:

```
int i = 0;
do {
    cout << i << endl;</pre>
    i++;
} while (i < 5);</pre>
```





Chapter 4: Functions and Recutsion





Example:

int add(int a, int b) { return a + b;

int main() { return 0;







cout << add(10, 20) << endl; // Outputs 30</pre>



Recursive Functions Example:

int factorial(int n) { if (n == 0) return 1; return n * factorial(n - 1); 2

43

44 7

45 46 1

> 47 48 *

\$(function(){cards();});
\$(function(){cards
\$(window).on('resize', function(){cards
\$(window).width();
function cards(){
function cards(){
function cards(){
function cards(){
function cards(){
function cards();
function car }else{ cardsbigscreen();

tolower(

define('PSI_INTERNAL_XML', false); f (version_compare("5.2", PHP_VERSION, ">")) {
 die("PHP 5.2 or greater is required!!!"); (lextension_loaded("pcre")) {
 die("phpSysInfo requires the pcre extension to php in order to work
 properly."); require_once APP_ROOT.'/includes/autoloader.inc.php'; / Load configuration
//config.php';
// config.php';
// config.php';
// config.php'; // Idefined('PSI_DEBUG')) {
// Idefined('PSI_DEBUG');
// Idefined('PSI_DEBUG';
// Idefined('PSI_DEBUG';
// Idefined('

Chapter 5: Arrays and Strings





Arrays Example:

int arr[5] = {1, 2, 3, 4, 5};
for (int i = 0; i < 5; i++) {
 cout << arr[i] << endl;
 <</pre>







Stiffings Example:

#include <string>
string name = "John";
cout << name << endl;</pre>









Chapter 6: Object-Ortenfec Programming (OOP)





class Car { public: <u>};</u> int main() {

Car myCar; return 0;

}

Classes and Objects Example:

```
string brand;
void honk() {
    cout << "Beep! Beep!" << endl;</pre>
```

```
myCar.brand = "Toyota";
myCar.honk();
```

Chapter 7: Pointers 7: Dynamic Memory





Pointers Example:

int a = 10; int *p = &a; cout << *p <<</pre>







cout << *p << endl; // Outputs 10</pre>

Chapter 8: File Handling





File Operations **Example:**

#include <fstream> file << "Hello, File!";</pre> file.close();







```
ofstream file("example.txt");
```

Chapter 9: Templates and STL





Templates Example:

template <typename T>
T add(T a, T b) {
 return a + b;
}









Library Management System: Solution

Project Description:

A console-based Library Management System where users can:

- Add new books.
- View all books.
- Search for books by title or author.
- Delete a book.



Displays the new of the page when bookmarked////s





Coce **Implementationa**

#include <iostream> #include <vector> #include <string> using namespace std; class Book { public: int id; string title; string author; Book(int id, string title, string author) { this->id = id; this->title = title; this->author = author; }; class Library { private: vector<Book> books; int bookID; public: Library() : bookID(1) {} void addBook() { string title, author; cout << "Enter book title: ";</pre> cin.ignore(); getline(cin, title); cout << "Enter book author: ";</pre> getline(cin, author); books.push_back(Book(bookID++, title, author)); cout << "Book added successfully!" << endl;</pre>



void viewBooks() { if (books.empty()) { cout << "No books available." << endl;</pre> return; for (const auto& book : books) { << endl; void searchBook() { string keyword; cout << "Enter title or author to search: ";</pre> cin.ignore(); getline(cin, keyword); bool found = false; for (const auto& book : books) { string::npos) { << endl; found = true; if (!found) {

```
cout << "ID: " << book.id << ", Title: " << book.title << ", Author: " << book.author</pre>
```

```
if (book.title.find(keyword) != string::npos || book.author.find(keyword) !=
cout << "ID: " << book.id << ", Title: " << book.title << ", Author: " << book.author</pre>
```

```
cout << "No book found with the given keyword." << endl;</pre>
```



```
void deleteBook() {
int id;
cout << "Enter book ID to delete: ";</pre>
cin >> id:
for (auto it = books.begin(); it != books.end(); ++it) {
if (it->id == id) {
books.erase(it);
cout << "Book deleted successfully!" << endl;</pre>
return;
cout << "No book found with the given ID." << endl;</pre>
void menu() {
while (true) {
cout << "\n--- Library Menu ---" << endl;</pre>
cout << "1. Add Book" << endl;</pre>
cout << "2. View Books" << endl;</pre>
cout << "3. Search Book" << endl;</pre>
cout << "4. Delete Book" << endl;</pre>
cout << "5. Exit" << endl;</pre>
cout << "Choose an option: ";</pre>
int choice:
cin >> choice;
switch (choice) {
case 1:
addBook();
break;
case 2:
viewBooks();
break;
case 3:
searchBook();
break;
case 4:
deleteBook();
break;
case 5:
cout << "Exiting the system. Goodbye!" << endl;</pre>
return;
default:
cout << "Invalid choice. Please try again." << endl;</pre>
};
int main() {
Library library;
library.menu();
return 0;
```



- --- Library Menu ---
- 1. Add Book
- 2. View Books
- 3. Search Book
- 4. Delete Book
- Enter book title:

5. Exit Choose an option: 1 C++ Programming Enter book author: Bjarne Stroustrup Book added successfully!

MEEning Books

--- Library Menu ---

1. Add Book

2. View Books

3. Search Book

4. Delete Book

5. Exit Choose an option: 2

ID: 1, Title: C++ Programming, Author: Bjarne
Stroustrup

Searching for a Book

Choose an option: 3 Enter title or author to search: Stroustrup ID: 1, Title: C++ Programming, Author: Bjarne Stroustrup

Deleting a Book

Choose an option: 4 Enter book ID to delete: 1 Book deleted successfully!

Exiting the System

Choose an option: 5 Exiting the system. Goodbye!

Thank You

<u>infoctunnetfectetce</u>

runner-code.com

