

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.Collections and LINQ
8.File Handling
9.Advanced Concepts
10.Advanced Projects





Defines the title in the browser bar(/):> Deprovides a name for the page when bookmarked(/):> Doisplays the name of the





C# (pronounced "C-sharp") is a modern, object-oriented programming language developed by Microsoft in the early 2000s as part of its .NET framework. Designed by Anders Hejlsberg, C# combines the power of C++ with the simplicity of Visual Basic, making it a versatile and easy-to-learn language for developers.

The formet and the second seco

- Key Features of C#:
 - reusable code.
- 2. Type-Safe:
- 4. Rich Standard Library:
- 5.Garbage Collection:
 - - applications.
- 8. Asynchronous Programming:

 - applications.

1.Object-Oriented Programming (OOP): • Supports principles like encapsulation, inheritance, polymorphism, and abstraction for building modular and

• Ensures robust error checking and eliminates common programming mistakes by enforcing strong typing. 3. Cross-Platform Development:

• With the advent of .NET Core and .NET 5+, C# allows developers to create applications for Windows, macOS, Linux, Android, and iOS.

• Offers an extensive library for handling tasks like file operations, database connectivity, and XML parsing.

• Built-in memory management reduces the risk of memory leaks and makes it easier to write efficient

6. Integration with .NET Framework:

• C# seamlessly integrates with the .NET ecosystem, supporting web, desktop, cloud, and mobile applications.

7.LINQ (Language Integrated Query):

• Provides a powerful syntax for querying data from databases, XML, collections, and more.

• Built-in support for asynchronous operations using the async and await keywords enhances responsiveness in

NHY LEAFN CHR





- Windows Development: C# is the primary language for developing Windows applications.
- Web Applications: Essential for building dynamic web applications using ASP.NET.
- Game Development: The preferred language for Unity, one of the most popular game engines.
- Enterprise Solutions: Widely used in large-scale enterprise software due to its scalability and reliability.

C# is known for its clean and readable syntax, making it an excellent choice for beginners and professionals alike. Whether you're building web apps, desktop apps, games, or cloud services, C# is a powerful and flexible language that can adapt to your development needs.



Setting Up the Development Environment



Studio Code. 2. Install the .NET SDK.







1. Download and install Visual Studio or Visual

3.Write and run your first program.



Your First CH Program

Code Examples



```
using System;
class Program
    static void Main()
        Console.WriteLine("Welcome to C#
Programming!");
ک
```



- Explanation:
 - operations.
 - console.

• using System;: Imports the System namespace for basic input/output • Console.WriteLine: Prints text to the

• Main: The entry point of the application.

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

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

int age = 25;float height = 5.9f; char grade = 'A'; bool isStudent = true; string name = "John";

Input and Output Operations Code Example:

using System; class Program

```
static void Main()
```

```
Console.Write("Enter your age: ");
int age = int.Parse(Console.ReadLine());
Console.WriteLine("You entered: " + age);
```

Operators



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

Examples

int a = 10, b = 20; Console.WriteLine(a + b); // Outputs 30 Console.WriteLine(a > b); // Outputs False







Chapter 3: Control Statements





if (age >= 18) ζ else







Console.WriteLine("You are an adult.");

Console.WriteLine("You are a minor.");



Switch Case Example:



```
switch (grade)
{
    case 'A':
        Console.WriteLine("Excellent!");
        break;
    case 'B':
        Console.WriteLine("Good!");
        break;
    default:
        Console.WriteLine("Try harder!");
        break;
}
```











For Loop Example:

for (int i = 0; i < 5; i++)</pre> Ł Console.WriteLine(i); }







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



While Loop Example:

```
int i = 0;
while (i < 5)
{
    Console.WriteLine(i);
    i++;
}</pre>
```





Do-Whfile Loop Example:

```
int i = 0;
do
Ł
    Console.WriteLine(i);
    i++;
} while (i < 5);</pre>
```





Chapter 4: Functions and Recursion





Defining Functions Example:

int Add(int a, int b)
{
 return a + b;
}
class Program
{
 static void Main()
 {
 Console.WriteLine(Add(10, 20)); //
Outputs 30
 }
}







Example:

Recursive Functions

```
int Factorial(int n)
Ł
    if (n == 0) return 1;
    return n * Factorial(n - 1);
ζ
```

42

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!!!"); (!extension_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
/ configuration
//config.php';
//config.php'; // !defined('PSI_DEBUG')) {
// !defined('PSI_DEBUG'));

Chapter 5: Arrays and Strings





Arrays **Example:**

int[] arr = {1, 2, 3, 4, 5}; foreach (int num in arr) Console.WriteLine(num);





Strings Example:

string name = "John"; Console.WriteLine(name);







Chapter 6: Object-Ottentec Programming (OOP)





class Car class Program };

Classes and Objects Example:

- public string Brand { get; set; }
- public void Honk()
 - Console.WriteLine("Beep! Beep!");

```
static void Main()
```

```
Car myCar = new Car { Brand = "Toyota"
myCar.Honk();
```

Chapter 7: Collections and LINO

LINO Example:


```
var evenNumbers = numbers.Where(n => n % 2 ==
0);
foreach (var num in evenNumbers)
٢
    Console.WriteLine(num);
3
```

Collections **Example:**

5 }; numbers.Add(6); foreach (int num in numbers) Console.WriteLine(num);







List<int> numbers = new List<int> { 1, 2, 3, 4,

Chapter 8: File Handling





File Operations **Example:**

using System.IO;

File.WriteAllText("example.txt", "Hello, File!"); string content = File.ReadAllText("example.txt"); Console.WriteLine(content);







Chapter 9: Advanced Concepts





Asynchronous Programming Example:

async Task FetchData()
{
 await Task.Delay(1000);
 Console.WriteLine("Data fetched.");
}







Final Project: Library Management System in C#

• This is a comprehensive implementation of a Library Management System in C#. The system allows users to manage books, add/remove books, and display the library catalog.





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





```
using System;
    using System.Collections.Generic;
 3
    namespace LibraryManagementSystem
 4
 5
         class Program
 6
 7
             // Book class to store book details
 8
             class Book
 9
10
                 public int ID { get; set; }
11
12
                 public string Title { get; set; }
13
                 public string Author { get; set; }
14
15
                 public Book(int id, string title, string author)
16
                    ID = id;
17
                                                        Code
                     Title = title;
18
19
                    Author = author;
20
21
22
                 public override string ToString()
23
24
                    return $"ID: {ID}, Title: {Title}, Author: {Author}";
25
26
```









```
10
11
12
13
14
15
17
18
19
21
22
23
24
25
27
31
32
33
34
35
36,
37
41
42
```

```
// Library class to manage books
class Library
    private List<Book> books = new List<Book>();
    public void AddBook(Book book)
        books.Add(book);
        Console.WriteLine($"Book '{book.Title}' added successfully.");
    public void RemoveBook(int id)
        Book bookToRemove = books.Find(book => book.ID == id);
        if (bookToRemove != null)
            books.Remove(bookToRemove);
            Console.WriteLine($"Book '{bookToRemove.Title}' removed successfully.");
        else
            Console.WriteLine("Book not found.");
    public void DisplayBooks()
        if (books.Count == 0)
            Console.WriteLine("No books in the library.");
        else
            Console.WriteLine("Library Catalog:");
            foreach (Book book in books)
                Console.WriteLine(book);
```

Code









```
11
12
13
15
21
22
23
41
42
47
52
```

```
static void Main(string[] args)
```

```
Library library = new Library();
bool exit = false;
```

while (!exit)

```
Console.WriteLine("\nLibrary Management System");
Console.WriteLine("1. Add Book");
Console.WriteLine("2. Remove Book");
Console.WriteLine("3. Display Books");
Console.WriteLine("4. Exit");
Console.Write("Choose an option: ");
string choice = Console.ReadLine();
```

```
switch (choice)
```

```
case "1":
   Console.Write("Enter Book ID: ");
   int id = int.Parse(Console.ReadLine());
   Console.Write("Enter Book Title: ");
   string title = Console.ReadLine();
   Console.Write("Enter Book Author: ");
   string author = Console.ReadLine();
    library.AddBook(new Book(id, title, author));
   break;
```

case "2":

```
Console.Write("Enter Book ID to Remove: ");
int removeId = int.Parse(Console.ReadLine());
library.RemoveBook(removeId);
break;
```

case "3":

```
library.DisplayBooks();
break;
```

case "4":

```
exit = true;
Console.WriteLine("Exiting the Library Management System.");
break;
```

default:

Console.WriteLine("Invalid option. Please try again."); break;



Code Implementation





Library Management System 1. Add Book 2. Remove Book Display Books 4. Exit Choose an option: 1 Enter Book ID: 101 Enter Book Title: C# Programming Enter Book Author: John Doe Book 'C# Programming' added successfully. 11 12 Library Management System 1. Add Book 13 2. Remove Book 15 Display Books 4. Exit 17 Choose an option: 1 Enter Book ID: 102 19 Enter Book Title: Advanced C# Enter Book Author: Jane Smith 21 Book 'Advanced C#' added successfully. 22 23 Library Management System 24 1. Add Book 2. Remove Book 25 Display Books 27 4. Exit Choose an option: 3 Library Catalog: ID: 101, Title: C# Programming, Author: John Doe ID: 102, Title: Advanced C#, Author: Jane Smith 32 Library Management System Add Book 2. Remove Book Display Books 4. Exit 37 -38 Choose an option: 2 Enter Book ID to Remove: 101 Book 'C# Programming' removed successfully. 41 Library Management System 42 43 1. Add Book 2. Remove Book Display Books 4. Exit 47 Choose an option: 4 48 Exiting the Library Management System.



How to Run The Program

- Copy the code into a C# IDE or text editor (e.g., Visual Studio, JetBrains Rider, or VS Code with C# extension).
- Compile and run the program.
- Follow the on-screen instructions to interact with the system.











Thank You

<u>infoctunnetfoccesco</u>

runner-code.com

