

Table of Contents



- [1. Structure of a C++ Program](#)
- [2. Basic Syntax and Data Types](#)
 - [Variables and Data Types](#)
 - [Constants](#)
- [3. Input/Output \(I/O\)](#)
- [4. Operators](#)
- [5. Conditional Statements](#)
 - [Else If Example](#)
- [6. Loops](#)
 - [For Loop](#)
 - [While Loop](#)
 - [Do-While Loop](#)
- [7. Functions](#)
 - [Function with Default Parameters](#)
- [8. Arrays](#)
- [9. Strings](#)
- [10. Pointers](#)
- [11. Classes and Objects](#)
- [12. Constructors](#)
- [13. Inheritance](#)
- [14. Exception Handling](#)
- [15. File I/O](#)
 - [Writing to a File](#)
 - [Reading from a File](#)
- [16. Useful STL \(Standard Template Library\) Containers](#)
 - [Vectors \(Dynamic Arrays\)](#)
 - [Maps \(Key-Value Pairs\)](#)

1. Structure of a C++ Program

```
#include <iostream> // Header for input/output
```

```
using namespace std;
```

```
int main() {  
    cout << "Hello, World!" << endl; // Print to console  
    return 0;  
}
```

2. Basic Syntax and Data Types

Variables and Data Types

```
int age = 25;           // Integer  
float height = 5.9;     // Floating point  
double pi = 3.14159;    // Double precision float  
char grade = 'A';       // Character  
bool isStudent = true;  // Boolean (true/false)  
string name = "Alice";  // String (requires <string> header)
```

Constants

```
const double GRAVITY = 9.8; // Unchangeable variable
```

3. Input/Output (I/O)

```
#include <iostream>
```

```
int main() {  
    string name;  
    cout << "Enter your name: ";  
    cin >> name;           // Input  
    cout << "Hello, " << name << endl; // Output  
    return 0;  
}
```

4. Operators

```
int a = 10, b = 3;

int sum = a + b;      // Addition
int diff = a - b;     // Subtraction
int prod = a * b;     // Multiplication
float div = a / (float)b; // Division (type casting)
int mod = a % b;      // Modulus

a++; // Increment
b--; // Decrement
```

5. Conditional Statements

```
int age = 20;

if (age >= 18) {
    cout << "Adult" << endl;
} else {
    cout << "Minor" << endl;
}
```

Else If Example

```
int score = 85;

if (score >= 90) {
    cout << "Grade A" << endl;
} else if (score >= 75) {
    cout << "Grade B" << endl;
} else {
```

```
    cout << "Grade C" << endl;  
}
```

6. Loops

For Loop

```
for (int i = 1; i <= 5; i++) {  
    cout << i << endl;  
}
```

While Loop

```
int count = 1;  
  
while (count <= 5) {  
    cout << count << endl;  
    count++;  
}
```

Do-While Loop

```
int n = 1;  
  
do {  
    cout << n << endl;  
    n++;  
} while (n <= 5);
```

7. Functions

```
#include <iostream>

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

int main() {
    cout << "Sum: " << add(10, 5) << endl;
    return 0;
}
```

Function with Default Parameters

```
int power(int base, int exponent = 2) {
    return pow(base, exponent);
}
```

8. Arrays

```
int numbers[5] = {1, 2, 3, 4, 5};

for (int i = 0; i < 5; i++) {
    cout << numbers[i] << endl;
}
```

9. Strings

```
#include <string>

string greet = "Hello";
```

```
greet += ", World!";  
cout << greet << endl;
```

10. Pointers

```
int value = 42;  
int* ptr = &value; // Pointer to value  
  
cout << "Address: " << ptr << endl; // Print address  
cout << "Value: " << *ptr << endl; // Dereference pointer
```

11. Classes and Objects

```
class Dog {  
public:  
    string name;  
    void bark() {  
        cout << name << " says Woof!" << endl;  
    }  
};  
  
int main() {  
    Dog myDog;  
    myDog.name = "Buddy";  
    myDog.bark();  
    return 0;  
}
```

12. Constructors

```
class Car {  
public:
```

```
    string brand;

    // Constructor
    Car(string b) {
        brand = b;
    }
};

int main() {
    Car car1("Toyota");
    cout << "Brand: " << car1.brand << endl;
    return 0;
}
```

13. Inheritance

```
class Animal {
public:
    void eat() {
        cout << "This animal eats food." << endl;
    }
};

class Cat : public Animal {
public:
    void meow() {
        cout << "Cat says Meow!" << endl;
    }
};

int main() {
    Cat kitty;
    kitty.eat();
    kitty.meow();
    return 0;
}
```

14. Exception Handling

```
try {  
    int x = 10 / 0;  
} catch (...) {  
    cout << "Error: Division by zero" << endl;  
}
```

15. File I/O

Writing to a File

```
#include <fstream>  
  
ofstream file("example.txt");  
file << "Hello, File!";  
file.close();
```

Reading from a File

```
#include <fstream>  
  
ifstream file("example.txt");  
string line;  
while (getline(file, line)) {  
    cout << line << endl;  
}  
file.close();
```

16. Useful STL (Standard Template Library) Containers

Vectors (Dynamic Arrays)

```
#include <vector>

vector<int> nums = {1, 2, 3};
nums.push_back(4);

for (int n : nums) {
    cout << n << endl;
}
```

Maps (Key-Value Pairs)

```
#include <map>

map<string, int> scores;
scores["Alice"] = 90;
scores["Bob"] = 85;

for (auto it : scores) {
    cout << it.first << ": " << it.second << endl;
}
```

This cheat sheet covers the basics of C++ to get you started.