C++ Programming Basics Cheat Sheet

Basic Structure of a C++ Program

```cpp
#include <iostream>

int main() {
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

Comments

Single-Line Comments

```cpp
// This is a single-line comment
int x = 10;  // This is an inline comment
```

Multi-Line Comments

```cpp
/*
This is a multi-line comment
spanning multiple lines.
*/
int y = 20;
```

Control Flow

If Statements

```cpp
if (condition) {
    // code block
} else if (anotherCondition) {
    // another code block
} else {
    // another code block
}
```

Switch Statement

```cpp
int day = 2;
switch (day) {
    case 1:
        std::cout << "Monday" << std::endl;
        break;
    case 2:
        std::cout << "Tuesday" << std::endl;
        break;
    // other cases
    default:
        std::cout << "Invalid day" << std::endl;
}
```

Variables and Data Types

Variables

```cpp
int number = 10;           // Integer
double pi = 3.14;          // Double
char letter = 'A';         // Character
std::string greeting = "Hello"; // String (requires <string> header)
bool isTrue = true;        // Boolean
```

Data Types

```cpp
int i = 100;               // Integer
float f = 10.5f;           // Floating point
double d = 10.5;           // Double floating point
char c = 'A';              // Character
bool b = true;             // Boolean
std::string s = "Hello";   // String (requires <string> header)
```

Operators

Arithmetic Operators

```cpp
+   // Addition
-   // Subtraction
*   // Multiplication
/   // Division
%   // Modulus
++  // Increment
--  // Decrement
```

Comparison Operators

```cpp
==  // Equal to
!=  // Not equal to
>   // Greater than
<   // Less than
>=  // Greater than or equal to
<=  // Less than or equal to
```

Logical Operators

```cpp
&&  // Logical AND
||  // Logical OR
!   // Logical NOT
```

Arrays

Defining and Using Arrays

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

// Access elements
std::cout << numbers[0] << std::endl;  // Output: 1

// Loop through array
for (int i = 0; i < 5; i++) {
    std::cout << numbers[i] << std::endl;
}

// Using range-based for loop (C++11 and later)
for (int number : numbers) {
    std::cout << number << std::endl;
}
```

Loops

For Loop

cpp Copy code

```cpp
for (int i = 0; i < 5; i++) {
    std::cout << i << std::endl;
}
```

While Loop

cpp Copy code

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

Do-While Loop

cpp Copy code

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

File Operations

Reading from a File

cpp Copy code

```cpp
#include <iostream>
#include <fstream>
#include <string>

int main() {
    std::ifstream file("filename.txt");
    std::string line;
    if (file.is_open()) {
        while (getline(file, line)) {
            std::cout << line << std::endl;
        }
        file.close();
    } else {
        std::cout << "Unable to open file";
    }
    return 0;
}
```

Writing to a File

cpp Copy code

```cpp
#include <iostream>
#include <fstream>

int main() {
    std::ofstream file("filename.txt");
    if (file.is_open()) {
        file << "Hello, file!\n";
        file.close();
    } else {
        std::cout << "Unable to open file";
    }
    return 0;
}
```

Functions

Defining and Calling Functions

cpp Copy code

```cpp
#include <iostream>

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

int main() {
    // Function call
    int result = add(5, 3);
    std::cout << result << std::endl;  // Output: 8
    return 0;
}
```

Function Declaration (Prototype)

cpp Copy code

```cpp
#include <iostream>

// Function declaration
int add(int, int);

int main() {
    int result = add(5, 3);
    std::cout << result << std::endl;  // Output: 8
    return 0;
}

// Function definition
int add(int a, int b) {
    return a + b;
}
```

Function Overloading

cpp Copy code

```cpp
#include <iostream>

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

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

int main() {
    std::cout << add(5, 3) << std::endl;       // Output: 8
    std::cout << add(5.5, 3.3) << std::endl;   // Output: 8.8
    return 0;
}
```

String Operations

Common String Methods

cpp Copy code

```cpp
#include <iostream>
#include <string>

int main() {
    std::string s = "Hello, World!";
    std::cout << s.length() << std::endl;        // Output: 13
    std::cout << s.substr(7, 5) << std::endl;    // Output: World
    std::cout << s.find("World") << std::endl;   // Output: 7

    std::string s2 = s.replace(7, 5, "C++");
    std::cout << s2 << std::endl;                // Output: Hello, C++!

    return 0;
}
```

Exception Handling

Try-Catch Block

cpp Copy code

```cpp
#include <iostream>
#include <stdexcept>

int main() {
    try {
        int result = 10 / 0;
    } catch (const std::exception &e) {
        std::cerr << "Exception: " << e.what() << std::endl;
    } catch (...) {
        std::cerr << "Unknown exception!" << std::endl;
    }

    return 0;
}
```