

Appendix-B-C++Reference-Guide

Appendix

B. C++ Syntax Reference Guide

Name	Syntax	Example
Pre-processor directives	<code>#include <LibraryName> #define <MACRO_NAME> <expansion></code>	...

`#include // for CIN & COUT #include // for setw() #include // for IFSTREAM #include // for STRING #include // for ISUPPER #include // for STRLEN #include // for ATOF #define PI 3.14159 #define LANGUAGE "C++"`

```

| Function | ``<ReturnType> <functionName>(<params>) {    <statements>    :
| Function parameters | ``<DataType> <passByValueVariable>, <DataType> & <pa
| COUT | cout << <expression> << ... ; | ``cout << "Text in quotes"    << 6
| Formatting output for money | cout.setf(ios::fixed); cout.setf(ios::showpoi
| Declaring variables | ``<DataType> <variableName>; <DataType> <variableName
| CIN | cin >> <variableName>; | cin >> variableName; |
| IF statement | ``if (<Boolean-expression>) {    <statements> } else {
| Asserts | assert(<Boolean-expression>); | ``#include <cassert> // at top of

```

```

| Name | Syntax | Example |
| :--: | :--: | :--: |
| FOR <br> loop | for (<initialization statement>; <br> <Boolean-expression>;
| WHILE <br> loop | ``while (<Boolean-expression>) {    <statements> }`` |
| DO-WHILE <br> Loop | do <br> \{ <statements> <br> \} <br> while (<Boolean-e
| Read from File | ``ifstream <streamVar>(<fileName>); if (<streamVar>.fail(
| Write to File | ``ofstream <streamVar>(<fileName>); if (<streamVar>.fail()
| Fill an array | ``<BaseType> <arrayName>[<size>]; <BaseType> <arrayName>[]
| C-Strings | char <stringName>[<size>]; <br> cin >> <stringName>; <br> for (
| String Class | ``string <stringName>; cin >> <stringName>; cout << <stringl

```

```

| Name | Syntax | Example |
| :--: | :--: | :--: |
| Switch | switch (<integer-expression>) <br> \{ case <integer-constant>: <br>
| Conditional Expression | <Boolean-expression> ? <expression> : <expression>
| Multidimensional array | <BaseType> <arrayName>[<SIZE>][<SIZE>]; <BaseType>
| Allocate memory | <ptr> = new(nothrow) <DataType>; <ptr> = new(nothrow) <Da
| Free memory | delete <pointer>; // one value delete [] <arrayPointer>; // al
| Command line parameters | ``int main(int <countVariable>, char **<arrayVar:
| Input errors | <istream>.fail(); <istream>.clear(); <istream>.ignore(<numCh:
| Exceptions | try <br> \{ <statements> throw <expression> \} catch (<declara

```

```

| Structures | struct <DataType> \{ <member variable declarations> \}; | ``s
| :--: | :--: | :--: |
| Header files | ``#ifndef <FILE_TAG> #define <FILE_TAG> <body of the header:
| makefile | <target> : <Dependency List> <Recipe> | ``a.out: file.o interfac
| Default parameters | ``<Return> <funcName>(<DataType> <variable> = <va
| Function pointer | <Return> (*<pointer>)(<param List>); | void (*ptrFunction
| Class syntax | ``class <ClassTag> {    <public/private>:    <variable
| Method definition | ``<Return> <ClassName>::<methodName>() {    <statemen

```