

C++ CHEAT SHEET

presented by the Student Computing Association

STANDARD SYSTEM TYPES & RANGE

int8_t	-128 to 127
uint8_t	0 to 255
int16_t	-32768 to 32767
uint16_t	0 to 65535
int32_t	-2147483648 to 2147483647
uint32_t	0 to 4294967295
int64_t	-9.2×10 ¹⁸ to 9.2×10 ¹⁸
uint64_t	0 to 1.8×10 ¹⁹

LITERALS & CO.

255	Integer
0xaf	Hexadecimal Integer
1234.0	double
234.243f	float
true	bool
"Hello World"	string / c-string
'a'	char

PREPROCESSOR

#include <LIB>	Includes standard header file LIB
#include "H.h"	Includes Header.h file
#define PI 3.14	Defines easy macro
#define f(a,b) a+b	More complex macro
#undef PI	Remove definition
#ifdef PI	Checks for definition; if true, code will compile
#else	Else part of the above if-statement
#endif	Ends the if-statement

NULLPOINTER

```
int* pInt;
//old, deprecated: if(pInt == NULL)
//new: if(pInt == nullptr)
//still valid: if(!pInt)
```

BITWISE OPERATORS

&	Bitwise AND
 	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
<<	Shift left
>>	Shift right

BOOLEAN LOGIC

==	Test of equality
!=	Test of non-equality
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal
!	NOT
&&	AND
 	OR

BASIC OPERATORS

+	addition
-	subtraction
*	multiplication
/	division
%	modulo
++var	increase before evaluation
var++	increase after evaluation
? :	short form of if-like structure
::	Scope Operator
->	pointer-to-member
·	Access member
<< >>	Bitshift / streams input/output

POINTERS: QUICK & DIRTY

&	Gets RAM address of variable (to save into pointer)
*	Dereferences pointer or defines a variable to be a pointer
->	Access pointer class member, same as (*pointer).member()
new	Create new object on heap, returns pointer
delete	Remove object at the pointer on heap

USING REPLACES TYPEDEF

```
//typedef is deprecated
typedef uint32_t uint32;
//now: using directive!
//using identifier = type_name;
using uint32 = uint32_t;
```

AUTO DATATYPE

```
//auto is an automatic datatype:
int x = 4; //equals:
auto y = 4;
//works for most cases, esp. STL:
for(std::vector<int>::iterator it = v.begin();...)
//with auto:
for(auto it = v.begin(); it != v.end(); ++it)
```

COMPILE-TIME ASSERTION

```
static_assert(bool_constexpr, string)
```

MULTITHREADING

```
//Thread Creation & Management:
void thread_func(int a) {
    std::cout << "My Number: " << a;
}
main() {
    std::thread t1(thread_func(1));
    t1.join();
}
```