C and C++ Cheat Sheet

libraries

```
#include input and output
functions
#include string related functions
#include memory allocation, rand,
and other functions
#include math functions
#include time related functions
```

functions

```
returnType functionName(
input1Type input1Name, input2Type
input2Name, …. )
{
// do something
return value; // value must be of
type returnType
}
```

comments

```
// one line comments this is a
C++ style one line comment
/* multiple line this is a
traditional C style comment
block comment */
```

variable types

char holds a character, or a number from -128 to 127 (1 byte)
bool holds a boolean value, either true or false (1 byte)
int hold an integer (a positive or negative number with NO decimal, 4 bytes)
float holds a real number (a positive or negative number with a decimal, 4 bytes)
void no type, raw binary data

conditionals

A == B if A is equal to B, this is true; otherwise, it's false
A != B if A is NOT equal to B, this is true; otherwise, it's false
A < B if A is less than B, this is true; otherwise, it's false A > B if A is greater B, this is true; otherwise, it's false
A <= B if A is less than or equal to B, this is true; otherwise, it's false A >= B if A is greater or equal to B, this is true; otherwise, it's false

control flow

```
if ( conditional )
{
// do something
}

if ( conditional )
{
// do something
}
else
{
// do something else
}
if ( conditional )
{
// do something
}
else if ( another_conditional )
{
// do something else
}
else
{
// do something as default
}
```

```
while ( conditional )
{
// do something
}
```

placing "break;" inside a while loop breaks out of the loop

placing "continue;" inside a while loop jumps to the start of the next loop

```
for ( initialization; test;
command )
{
// do something
}
```

"break;" and "continue;" can be used within for loops as well with identical effects

this is equivalent to:

```
initialization;
while( test )
{
// do something
command;
}
```

```
switch ( variable )
{
case value1:
// do something
break;
case value2:
// do something else
break;
default:
// do something by default
break;
}
```

this is equivalent to:

```
if ( variable == value1 )
{
// do something
}
else if ( variable = value2 )
{
// do something else
}
else
{
// do something by default
}
```

printf formats

%d: integer
%f: float or double
%s: string (char array)
%c: char (single character)

scanf formats

%d: integer
%f: float
%lf: double (first character is L, not one!)
%s: string (char array)
%c: char (single character)

string methods

```
/* to use these methods, you
must include */
strcpy(char dest[], char src[])
copies src into dest
int strlen(char s[])
returns length of s
int strcmp(char s1[], char s2[])
returns negative if s1 < s2, 0 if
s1 == s2 positive if s1 > s2
strcat(char dest[], char src[])
adds src to the end of dest
```

abstract classes and methods

```
virtual void sound(char s[]) = 0;
// Reminder: no "abstract"
keyword.
// Class headers do not indicate
// whether the class is abstract
or
// not. A class is abstract if it
// contains any abstract methods.
```

Source: https://goo.gl/wKqtSY