**Computational Form**
Jared Schiffman
Fall 2007

# C/C++ Cheat Sheet (v1)

**libraries**

```
#include <stdio.h>          input and output functions
#include <string.h>         string related functions
#include <stdlib.h>         memory allocation, rand, and other functions
#include <math.h>           math functions
#include <time.h>           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
}