

2012 ACM@UVa HSPC C++ Cheatsheet

Primitive Data Types

int	32-bit signed two's complement integer
float	32-bit floating point number
double	64-bit floating point number
bool	Data type with two possible values: true or false
char	8-bit ASCII character

Operations

+	Arithmetic addition or String concatenation
-	Arithmetic subtraction
/	Arithmetic division
%	Integer division remainder (modulus)
++	Increment
--	Decrement
==	Equality
!=	Inequality
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal
&&	Logical AND
!	Logical NOT
	Logical OR

Variable Declaration and Assignment

```
int      index      =      0;
TYPE      NAME      ASSIGNMENT      VALUE
```

If Statement

```
if ( Boolean Expression ){
    Statements;
}
```

While Loop

```
while ( Boolean Expression ){
    Statements;
}
```

For Loop

```
for ( Initialization ; Termination ; Increment){
    Statements;
}
```

Strings

```
#include <string>
string a = "UVa";
    Creates the string a with value "UVa".
string b = "HSPC";
    Creates the string b with value "HSPC".
boolean falseValue = a.compare(b) ;
    a does not have the same value as b.
char letterU = a[0];
    The first character of a is the letter "U".
int zero = a.find("U");
    The letter "U" is the first character in the string a.
int minusOne = a.find("X");
    The letter "X" does not appear in the string, returning -1.
string uvaHSPC = a + b;
    The newly created string is "UVAHSPC".
```

Arrays

```
int[]      array      =      new      int[size];
ARRAY TYPE      NAME      ARRAY LENGTH
```

```
array[index] = 50;
int fifty = array[index];
```

Function Declaration

```
int      factorial      (int n)
RETURN TYPE      METHOD NAME      ARGUMENTS
```

```
int factorial(int n){
    /*body*/
}
```

2012 ACM@UVa HSPC C++ Cheatsheet

Data Structures

Math

```
#include <math.h>
```

All return doubles. Angles are in radians.

<code>exp(1.0)</code>	The base of the natural logarithm.
<code>sin(ang)</code>	Computes the sine of <code>ang</code> .
<code>cos(ang)</code>	Computes the cosine of <code>ang</code> .
<code>tan(ang)</code>	Computes the tangent of <code>ang</code> .
<code>asin(ang)</code>	Computes the inverse sine of <code>ang</code> .
<code>log(a)</code>	The natural logarithm of <code>a</code> .
<code>sqrt(a)</code>	The square-root of <code>a</code> .
<code>pow(a,b)</code>	Raises <code>a</code> to the power of <code>b</code> .
<code>fabs(a)</code>	Returns the absolute value <code>a</code> .

Input

```
using namespace std;
#include <iostream>
```

```
cin >> declaredInt;
```

Reads an integer from standard input.

```
cin >> declaredString;
```

Reads a string from standard input.

```
cin >> declaredDouble;
```

Reads a double from standard input.

Output

```
cout << "Print the value : " << dog << endl;
```

Prints out a the string and the value of the variable `dog` with a new line.

Vector

```
using namespace std;
```

```
#include <vector>
```

```
vector<int> list(20);
```

Creates a new vector of integers .

```
list[0] = 1;
```

Assigns the first element of the list to 1.

```
cout << list[0];
```

Prints the first element of the list, the number 1.

Map

```
using namespace std;
```

```
#include <map>
```

```
map<string,string> dict;
```

Creates a mapping from strings to strings.

```
dict["Dog"] = "Cat";
```

Maps the string "Dog" (key) to "Cat" (value).

```
cout << dict["Dog"] << "\n";
```

Retrieves the value for the key "Dog" and prints the word "Cat".