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
-	Arithmetic subtraction
/	Arithmetic division
%	Integer division reminder (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.h>
char a[4] = "UVa";
```

Creates the string a with value "Uva". Array size must be one more than the string length.

```
char b[5] = "HSPC";
```

Creates the string b with value "HSPC". Array size must be one more than the string length.

```
int falseValue = !strcmp(a,b) ;
```

a does not have the same value as b.

```
char letterU = a[0];
```

The first character of a is the letter 'U'.

Arrays

```
   int     array     [10];
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*/
}
```

UVa HSPC C Cheatsheet

Math

`#include <math.h>`
All return doubles. Angles are in radians.

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

Input

```
#include <stdio.h>
```

```
scanf ("%d", &declaredInt);
```
Reads an integer from standard input.
```
scanf ("%s", declaredCharArray);
```
Reads a string from standard input; note no ampersand!
```
scanf ("%lf", &declaredDouble);
```
Reads a double from standard input.

Output

```
printf ("Print the value: %d\n", dog);
```
Prints out a the string and the value of the variable dog with a new line.

Data Structures

No C libraries are included with provided data structures. The C++ Standard Template Library (STL) is provided with the C++ language.