**Primitive Data Types**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>32-bit signed two's complement integer</td>
</tr>
<tr>
<td>float</td>
<td>32-bit floating point number</td>
</tr>
<tr>
<td>double</td>
<td>64-bit floating point number</td>
</tr>
<tr>
<td>bool</td>
<td>Data type with two possible values: true or false</td>
</tr>
<tr>
<td>char</td>
<td>8-bit ASCII character</td>
</tr>
</tbody>
</table>

**Operations**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Arithmetic addition or String concatenation</td>
</tr>
<tr>
<td>-</td>
<td>Arithmetic subtraction</td>
</tr>
<tr>
<td>/</td>
<td>Arithmetic division</td>
</tr>
<tr>
<td>%</td>
<td>Integer division reminder (modulus)</td>
</tr>
<tr>
<td>++</td>
<td>Increment</td>
</tr>
<tr>
<td>--</td>
<td>Decrement</td>
</tr>
<tr>
<td>==</td>
<td>Equality</td>
</tr>
<tr>
<td>!=</td>
<td>Inequality</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>Logical AND</td>
</tr>
<tr>
<td>!</td>
<td>Logical NOT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**If Statement**

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

**While Loop**

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

**For Loop**

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

**Strings**

```cpp
#include <string>

string a = "UVa";
string b = "HSPC";

boolean falseValue = a.compare(b);

char letterU = a[0];
int zero = a.find("U");
int minusOne = a.find("X");

string uvaHSPC = a + b;
```

**Arrays**

```cpp
int[] array = new int[size];

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

**Function Declaration**

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

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.

Data Structures

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”.