C++ II

Fundamental Types, Constants, and Variables

Data Types

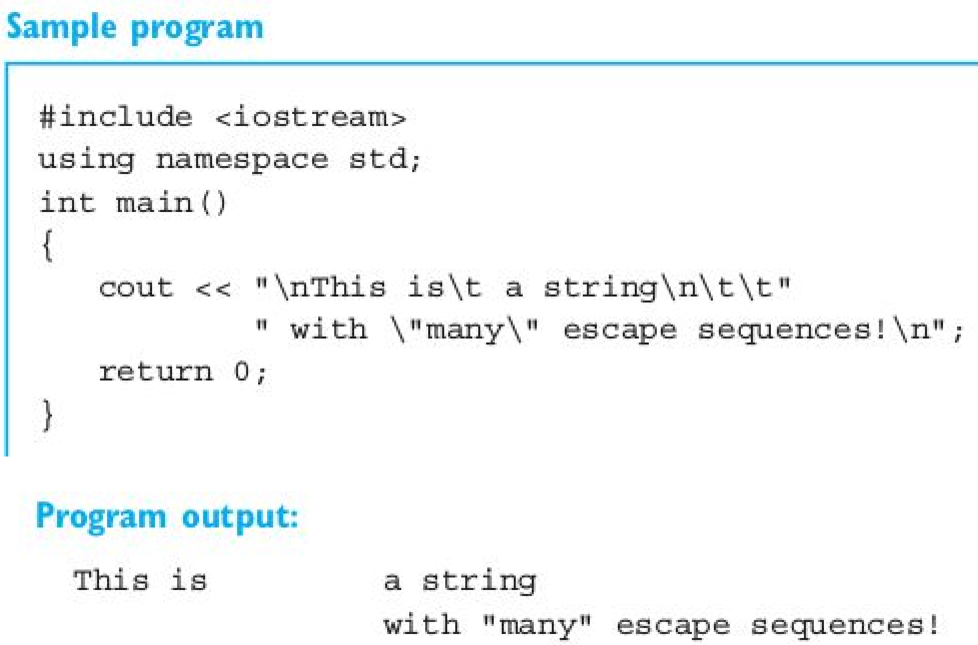
- C++ is a strictly typed language (like Java): It determines:
    - The internal representation of the data
    - The amount of memory to allocate
- Types:
    - bool
    - char, wchar_t
    - Arithmetic types:
        - Integral types: short, int, long
            - signed
            - unsigned
        - Floating point types: float, double, long double

- The sizeof operator: gives you the amount of memory to store an object in bytes
- void: expressions that do not represent a value
    - A function call can take a void type

Constants

- Literals are constants
  - Boolean constants: true, false
  - Numerical constants:
    - decimal: begins with a decimal number other than 0. Examples: `109` or `987650`
    - octal: begins with leading `0`. Examples: `077`
    - hexadecimal: begins with `0x` or `0X`. Examples: `0x2A0`
    - designate the type of a constant by adding letter `L, l, U, u`
    - Floating point numbers/literals are represented as decimals, cam also use exponential notation, double by default
  - Character constants
    - Denoted by single quotes around a character
  - String constants: a sequence of characters, stored without the quotes and terminated by /0

Control and Special Characters

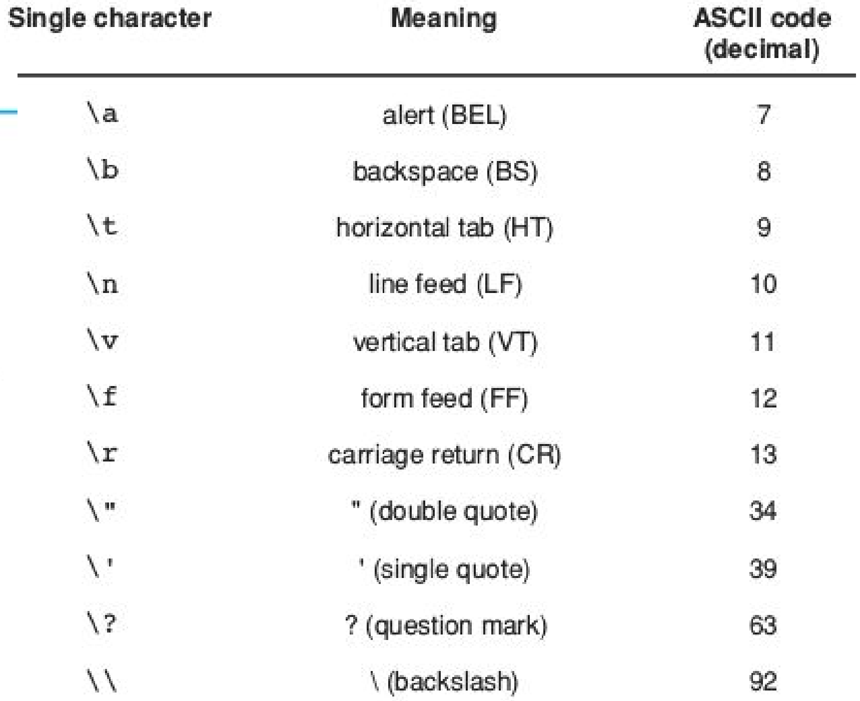


Sample program

```
#include <iostream>
using namespace std;
int main()
{
   cout << "\nThis is\t a string\n\t\t"
           " with \"many\" escape sequences!\n";
   return 0;
}
```

Program output:

```
This is          a string
                 with "many" escape sequences!
```



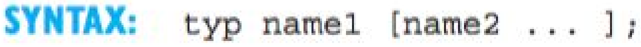
Single character	Meaning	ASCII code (decimal)
`\a`	alert (BEL)	7
`\b`	backspace (BS)	8
`\t`	horizontal tab (HT)	9
`\n`	line feed (LF)	10
`\v`	vertical tab (VT)	11
`\f`	form feed (FF)	12
`\r`	carriage return (CR)	13
`\"`	" (double quote)	34
`\'`	' (single quote)	39
`\?`	? (question mark)	63
`\\`	\ (backslash)	92

Valid Names for Variables or Functions

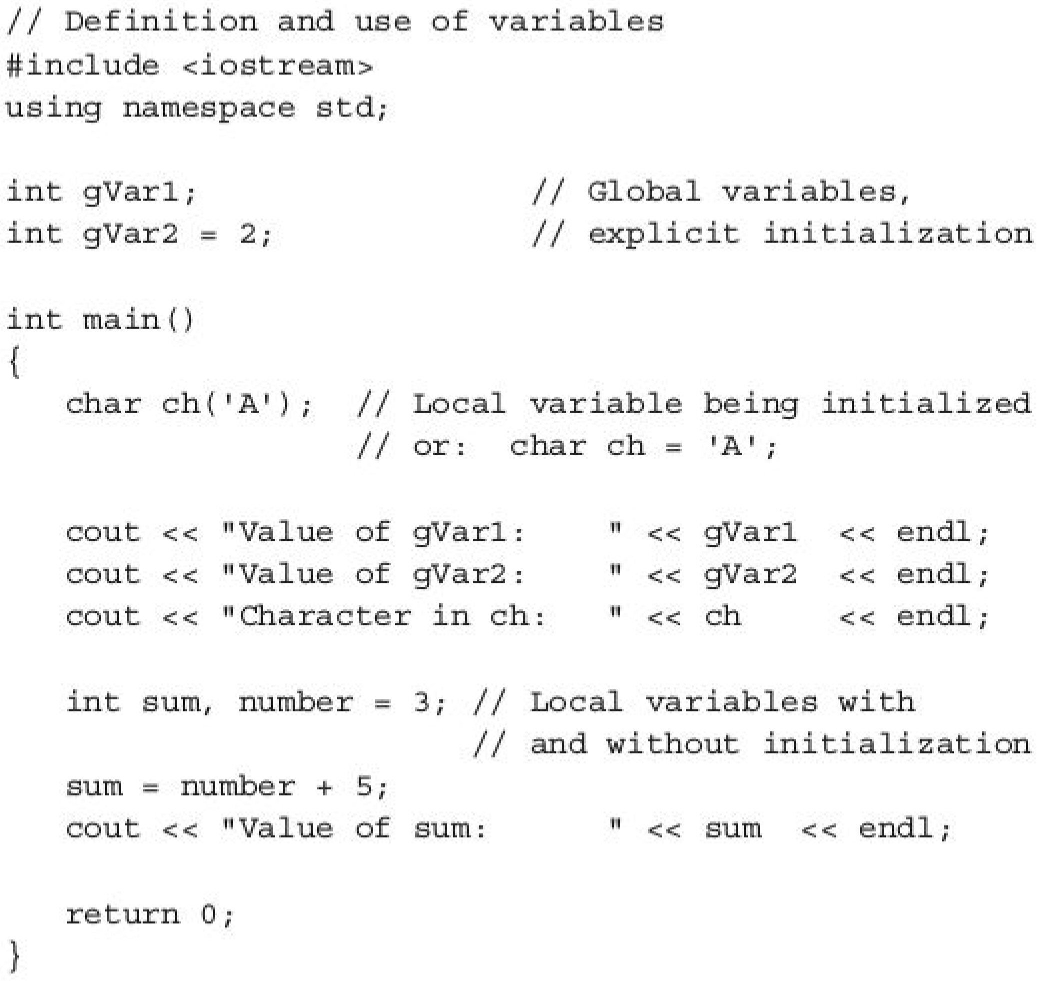
- Cannot be a keyword
- Must begin with a letter or underscore
- Conventions
  - c, ch for characters
  - i, j, k, l, m, n for integers, especially indices
  - x, y, z for floating point numbers

Variables

Examples of some global and local variables: global --- outside a function, local --- inside a function



SYNTAX: `typ name1 [name2 ... ];`



```
// Definition and use of variables
#include <iostream>
using namespace std;

int gVar1;                        // Global variables,
int gVar2 = 2;                    // explicit initialization

int main()
{
   char ch('A');  // Local variable being initialized
                  // or:  char ch = 'A';

   cout << "Value of gVar1:    " << gVar1  << endl;
   cout << "Value of gVar2:    " << gVar2  << endl;
   cout << "Character in ch:   " << ch     << endl;

   int sum, number = 3; // Local variables with
                        // and without initialization
   sum = number + 5;
   cout << "Value of sum:      " << sum  << endl;

   return 0;
}
```

Initialization, `const`, and `volatile`

- Initialization syntax:
  - `char c = 'a';`
  - `float x(1.875);`
- const: makes a variable read-only
  - `const double pi = 3.1415947;`
- volatile: can be modified by extra-program events:
  - Compiler continually checks value in memory (assumes it’s been changed)
  - Syntax: `volatile unsigned long clock_ticks;`
  - `const` and `volatile` can be combined:
    - `volatile const unsigned long time_to_live;`