VARIABLES
INPUT & OUTPUT
KEYWORDS
COMMENTS

IN C++ LANGAUGE

Setting By : Lect. Waleed Rasheed

Second Lecture

Variables :

A variable is a symbolic name for a memory location in which data can be stored and subsequently recalled. Variables are used for holding data values so that they can be utilized in various computations in a program. All variables have two important attributes:

· A **type** which is established when the variable is defined (e.g., integer, real, character). Once defined, the type of a C++ variable cannot be changed.

· A **value** which can be changed by assigning a new value to the variable. The kind of values a variable can assume depends on its type. For example, an integer variable can only take integer values (e.g., 2, 100, -12).

An **identifier** is a user defined name; variable names are identifiers. Identifiers must not be spelled the same as keywords such as if and while.

A name should consist of one or more characters, each of which may be a letter (i.e., 'A'-'Z' and 'a'-'z'), a digit (i.e., '0'-'9'), or an underscore character ('_'), except that the first character may not be a digit. Upper and lower case letters are distinct. For example:

salary // valid identifier
salary2 // valid identifier
2salary // invalid identifier (begins with a digit)
_salary // valid identifier
Salary // valid but distinct from salary

C++ Keywords

C++ reserves a set of words for use within the language as keywords. Keywords may not be used as program identifiers.

asm	do	if	return	try
auto	double	inline	short	typedef
bool	dynamic_cast	int	signed	typeid
break	else	long	sizeof	typename
case	enum	mutable	static	union
catch	explicit	namespace	static_cast	unsigned
char	export	new	struct	using
class	extern	operator	switch	virtual
const	false	private	template	void
const_cast	float	protected	this	volatile
continue	for	public	throw	wchar_t
default	friend	register	true	while
delete	goto	reinterpret_cast		

C++ defines a set of arithmetic types, which represent integers, floating-point numbers, and individual characters and boolean values. In addition, there is a special type named void. The void type has no associated values and can be used in only a limited set of circumstances. The void type is most often used as the return type for a function that has no return value.

The size of the arithmetic types varies across machines. By size, we mean the number of bits used to represent the type. The standard guarantees a minimum size for each of the arithmetic types, but it does not prevent compilers from using larger sizes. Indeed, almost all compilers use a larger size for int than is strictly required.

Type	Meaning	Minimum Size
char	character	8 bits
short	short integer	16 bits
int	integer	16 bits
long	long integer	32 bits
float	single-precision floating-point	6 significant digits
double	double-precision floating-point	10 significant digits
long double	extended-precision floating-point	10 significant digits
Char*	String	“ text “

Examples

```
int salary = 32767;  // -32769 <int<32767
long price = 4500000;
float interestRate = 0.06;
double pi = 3.141592654; //but rounded to 5 digit after dot.
char ch = 'A';
char *str = "HELLO";
```

Note:

A long string may extend beyond a single line, in which case each of the
preceding lines should be terminated by a backslash. For example:

```
"Example to show \
the use of backslash for \
writing a long string"
```

The backslash in this context means that the rest of the string is continued on the next line. The above string is equivalent to the single line string:

```
"Example to show the use of backslash for writing a long string"
```

Simple Input/Output

The most common way in which a program communicates with the outside world is through simple, character-oriented Input/Output (IO) operations. C++ provides two useful operators for this purpose: >> for input and << for output.

Examples

```
cin>>x>>y>>z;
cout<<x<<y<<z;
cout<<“sum=“<<sum;
cout<<“x+y=“<<x+y;
```

Comments

Comments start with /* and end with */. These comments do not read. For example:

```
/* this is a comment
 still a comment */
```

A comment can also start with //, extending to the end of the line. For example:

```
const int max_widget = 42; // Largest size of a widget
```

note:

```
// with one line …
/* with multiline … */
```

Escape Sequences for Nonprintable Characters

Some characters are nonprintable. A nonprintable character is a character for which there is no visible image, such as backspace or a control character. Other characters have special meaning in the language

newline	\n	horizontal tab	\t
vertical tab	\v	backspace	\b
carriage return	\r	formfeed	\f
alert (bell)	\a	backslash	\\
question mark	\?	single quote	\'
double quote	\"		

Arithmetic Operators

C++ provides five basic arithmetic operators.

Operator	Name	Example
+	Addition	12 + 4.9 // gives 16.9
-	Subtraction	3.98 - 4 // gives -0.02
*	Multiplication	2 * 3.4 // gives 6.8
/	Division	9 / 2.0 // gives 4.5
%	Remainder	13 % 3 // gives 1

Integer division always results in an integer outcome (i.e., the result is always rounded down). For example:

```
9 / 2 // gives 4, not 4.5!
-9 / 2 // gives -5, not -4!
```

To obtain a real division when both operands are integers, you should cast one of the operands to be real:

```
int cost = 100;
int volume = 80;
double unitPrice = cost / (double) volume; // gives 1.25
```