Introduction to programming Languages CS-103

Class: BS-Mathematics 3rd Semester

Rawalpindi Women University

By: Khaula Qadeer

Overview:

- To write meaningful programs, you must learn the programming language's special symbols, words, and syntax rules.
- The syntax rules tell you which statements (instructions) are legal or valid, that is, which are accepted by the programming language and which are not.
- You must also learn semantic rules, which determine the meaning of the instructions.

Comments

- Non executable statements in a program.
- Comments are for the reader, not for the compiler.

1. Single-line comments (//)

```
cout << "7 + 8 = " << 7 + 8 << endl; //prints: 7 + 8 = 15
```

1. Multiple-line comments (/* and */)

```
/*
        You can include comments that can
        occupy several lines.
*/
```

Special Symbols

- The smallest individual unit of a program written in any language is called a token.

1. **Special symbols:**

   + - * / . ; ? , <= != == >=

2. **Reserved Words (Keywords):**

   int, float, double, char, const, void, return

3. **Identifiers (Names of thing):**

   variables, constants, and functions

- consists of letters, digits, and the underscore character (_)
- must begin with a letter or underscore

Illegal Identifiers

Identifiers can be made of only letters, digits, and the underscore character; no other symbols are permitted to form an identifier.

Illegal Identifier	Description
`employee Salary`	There can be no space between `employee` and `Salary`.
`Hello!`	The exclamation mark cannot be used in an identifier.
`one+two`	The symbol + cannot be used in an identifier.
`2nd`	An identifier cannot begin with a digit.

NOTE

Compiler vendors usually begin certain identifiers with an underscore (_). When the linker links the object program with the system resources provided by the integrated development environment (IDE), certain errors could occur. Therefore, it is advisable that you should not begin identifiers in your program with an underscore (_).

Whitespaces

- Blanks, tabs, and newline characters.
- Use to separate special symbols, reserved words, and identifiers.
- Nonprintable
- Make the program more readable.

Bits and Bytes

- At the smallest scale in the computer, information is stored as bits and bytes.
- **Bit**
- a "bit" is atomic: the smallest unit of storage
- A bit stores just a 0 or 1
- "In the computer it's all 0's and 1's" ... bits
- Anything with two separate states can store 1 bit
- In a chip: electric charge = 0/1
- In a hard drive: spots of North/South magnetism = 0/1
- A bit is too small to be much use
- Group 8 bits together to make 1 byte

Byte

- 1 byte is group of 8 bits
- 8 bits can make 256 different patterns
- How to use the 256 patterns?
- How to store a number in a byte?
- Start with 0, go up, one pattern per number, until run out of patterns
- 0, 1, 2, 3, 4, 5, ... 254, 255
- One byte can hold a number between 0 and 255
- i.e. with 256 different patterns, we can store a number in the range 0..255
- Really good for storing characters/letters.

Byte

- "Byte" - unit of information storage
- A document, an image, a movie .. how many bytes?
- 1 byte is enough to hold about 1 typed character, e.g. 'b' or 'X' or '$'
- All storage is measured in bytes, despite being very different hardware
- **Kilobyte**, KB, about 1 thousand bytes
- **Megabyte**, MB, about 1 million bytes
- **Gigabyte**, GB, about 1 billion bytes
- **Terabyte**, TB, about 1 trillion bytes (rare)

Characters ASCII values

- ASCII (American Standard Code for Information Interchange) is an encoding representing each typed character by a number.
- Each number is stored in one byte (so the number is in 0..255)
- A is 65
- B is 66
- a is 96
- space is 32
- "Unicode" is an encoding for mandarin, greek, arabic, etc. languages, typically 2-bytes per "character"

ASCII Values of Characters

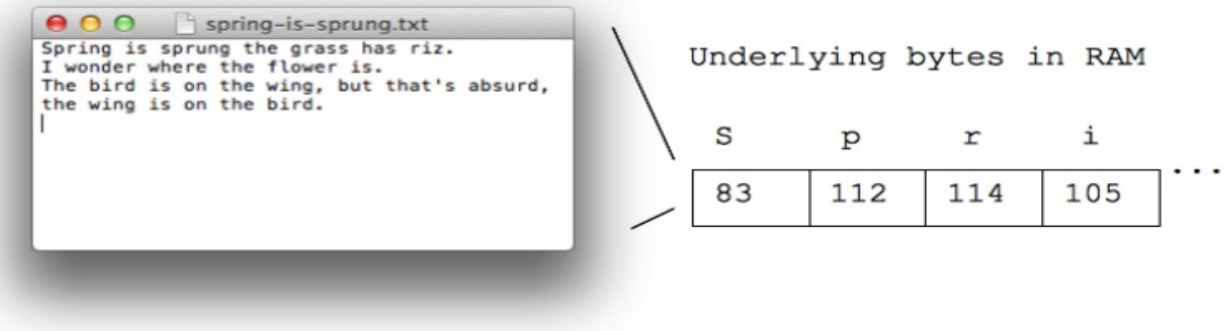
32	space
33	!
34	"
35	#
36	$
37	%
38	&
39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@

65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[
92	\
93	]
94	^
95	_
96	`

97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{
124	|
125	}
126	~

Typing, Bytes, and You

- Each letter is stored in a byte, as below
- 100 typed letters takes up 100 bytes
- When you send, say, a text message, the numbers are sent
- Text is quite compact, using few bytes, compared to images etc.



Numbers in Computers

- One byte works well for individual characters, but computers are also good at manipulating numbers.
- **Integers** are typically stored with either 4 or 8 bytes
  4 bytes can store numbers between -2147483648 and 2147483647
  8 bytes can store numbers between -9223372036854775808 and 9223372036854775807
- Adding in binary is just like normal addition with carrying
  But when you run out of bits you can't carry anymore
  Leftmost bit indicates sign, so carrying to the leftmost bit changes a number ffrom positive to negative.
  So adding 1 to 2147483647 goes to -2147483648!
  Called **Integer Overflow**
  Integer Overflow and Gangam Style

Data Types

1. **Simple data type:**

   Fundamental data type in C++

I. **Integral:** that deals with integers, or numbers without a decimal part.

II. **Floating-point:** deals with decimal numbers

III. **Enumeration:** user-defined data type

2. **Structured data type**
3. **Pointers**

Integral data classification

- Further Nine categories: char, short, int, long, bool, unsigned char, unsigned short, unsigned int, and unsigned long.
- We will learn only 5: integer, real, char, bool, and the enumeration type.

TABLE 2-2 Values and Memory Allocation for Three Simple Data Types

Data Type	Values	Storage (in bytes)
`int`	-2147483648 to 2147483647	4
`bool`	`true` and `false`	1
`char`	-128 to 127	1

int DATA TYPE

- in mathematics, these are numbers such as:

  -6728, -67, 0, 78, 36782, +763

- Two rules from these examples:

  1. Positive integers do not need a + sign in front of them.

  2. No commas are used within an integer. Recall that in C++, commas are used to separate items in a list. So 36,782 would be interpreted as

- two integers: 36 and 782.

bool DATA TYPE

- The data type bool has only two values: **true** & **false**.
- **true** and **false** are called the logical (Boolean) values.
- The central purpose of this data type is to manipulate logical
- (Boolean) expressions.
- Logical (Boolean) expressions will be formally defined and discussed

in detail later.

- In C++, bool, true, and false are reserved words.

char DATA TYPE

- **char** is the smallest integral data type.
- It represents single characters— letters, digits, and special symbols.
- enclose each character represented within single quotation marks.
- 'A', 'a', '0', '*', '+', '$', '&', ' ‘
- Several different character data sets are currently in use.

I. American Standard Code for Information Interchange (ASCII) . ASCII character set has 128 values.

II. Extended Binary- Coded Decimal Interchange Code (EBCDIC). The EBCDIC character set has 256 values and was created by IBM.

Floating-Point Data Types

- Decimal numbers

  43872918 = 4.3872918 * 107 {10 to the power of seven}

  .0000265 = 2.65 * 10-5 {10 to the power of minus five}

  47.9832 = 4.79832 * 101 {10 to the power of one

TABLE 2-3 Examples of Decimal Numbers in Scientific and C++ Floating-Point Notations

Decimal Number	Scientific Notation	C++ Floating-Point Notation
75.924	$7.5924 * 10^{1}$	7.592400E1
0.18	$1.8 * 10^{-1}$	1.800000E-1
0.0000453	$4.53 * 10^{-5}$	4.530000E-5
-1.482	$-1.482 * 10^{0}$	-1.482000E0
7800.0	$7.8 * 10^{3}$	7.800000E3

Floating-Point Data Types

- C++ provides three data types to manipulate decimal numbers: float, double, and long double.
- In newer compilers, the data types double and long double are the same Therefore, only the data types **float** and **double** are described here
- **float:** The data type float is used in C++ to represent any decimal number between -3.4 *10^38 and 3.4 *10^38. The memory allocated for a value of the float data type is four bytes.
- **double:** The data type double is used in C++ to represent any decimal number between -1.7 *10^308 and 1.7 *10^308. The memory allocated for a value of the double data type is eight bytes.

Types of errors in Programming Language

- Programming languages follow very strict conventions.

1. **Compile-time error/Syntax error:** is a violation of the programming language rules that is detected by the compiler.
2. **Run-time error:** causes a program to take an action that the programmer did not intend.
3. **Logic errors**: caused by logical flaws in the program. Complier doesn't detect this error. E.g: to open the file which doesn't exist. To perform illegal operation (1/0)