

CS221: Computer Programming I

B.Sc in Computer Science

Autumn 2011

CS221: Computer Programming I

Chapter II: Basic elements of C++

Basic elements of C++: “building blocks”

- Keywords, identifiers, operators
- Data types: Numbers and characters (string)
- Variables and constants
- (Predefined or built-in) functions
- Expressions
- Punctuation marks

Basic elements of C++: Basic Statements

- The null statement
- The block statement
- Assignment statement
- Input/Output statements
- Expression statements

Keywords

- **Keywords:** These are names for which the compiler has already attached a meaning. Also called reserved words, keywords constitute the vocabulary of the language.
- **Examples:** int, float, const, break, switch, case, char, if, while, else, for, do, struct, union, enum, ...

Identifiers

- **Identifiers:** An identifier is a name given to identify the following objects: Constants, Variables, Data types, Enumerated Elements and Functions names. Identifiers are almost always provided by the programmers.
- *Rules of Identifiers*
 1. A combination of letters, digits or underscore (_)
 2. First character can not be a digit
 3. Case sensitive
 4. _____
 5. _____

Identifiers

An identifier must be **declared** (introduced) before use.

Exc: What is declaration statement?

Operators

- Arithmetic operators: `+` `-` `*` `/` `%`
(modulus)
- Increment/Decrement: `++` `--`
- Logical operators: `!` `&&` `||`
- Relational operators: `==` `<` `<=` `>`
`>=` `!=`
- Bitwise logical operators: `~` `&` `|`
- Assignment operator: `=`

Reading Assignment

What is the output of $2+3*5$?

Describe the priority of operators in C++.

Data Types

- A data type is a collection of data elements of “similar” characteristics.
 - **Simple data types:** “Integers”, “Reals”, char, boolean, “pointer”, enum
 - **Structured data types:** “array”, “string”, struct, union

“Integers”: Numbers without decimal points

Name	Size	Range of Values	
		Signed	Unsigned
	1 byte	-128 --- +127	0 --- 255
short	2 bytes		
int			
long	4 bytes		

“Integers”

- Operators valid on “integers”:



- _____



- _____



- _____



- _____



- _____



- _____

“Reals”

Name	Size	Range of Values
float	4 bytes	
double	8 bytes	
long double	10 bytes	

“Reals”

- Operators valid on “reals”:



- _____



- _____



- _____

char

- Range of values: the 256 ASCII characters
- Size: _____
- Operators: _____

char

- Character constants are single characters enclosed within single quotes
- Example of valid char constants: 'A' '?' " '3'
- Example of invalid char constants: 3 "A"
- Note: Refer to the ASCII Table for the complete collection of char type.

Reading Assignment

ASCII vs Unicode

The char data type in C++

char in C++ is a 1-byte “integer” type. Refer to the “integers” table again...

“Integers”: Numbers without decimal points

Name	Size	Range of Values	
		Signed	Unsigned
char	1 byte		
short	2 bytes		
int			
long	4 bytes		

bool (also called boolean or logical)

- Range of values: *false true*
- Size: _____
- Operator: _____

The `enum` data type

Exc: Describe the `enum` type.

“String”

- Members: a sequence of (ASCII/Unicode) chars
- Size: $> \text{Number of characters} * 1 \text{ byte (per char)}$
- Operators: None
- String Constant: A sequence of characters enclosed within double quotes (“ ”)
- Example: “Hello HiLCoE”, “A”, “”, “2*5/ X”

Variables

- a high level language representation of memory locations.
- Variables are used to store data.

Variable identifier declaration

- Memories that are used in a program must be reserved using variable identifier declarations as follows:
- Syntax:
- **<type> <list of idf> ;**
- Description:
 - The list of identifiers are variable identifiers of the given type.
 - Reserve an appropriate size memory space for the list of (variable) identifiers.

Variable identifier declaration

- *Examples:*
- `float x;`
- `long double height, bekele, cost1;`
- `char yourAnswer; // character variable`
- `char student_name[20]; // string variable`
- *Exercise:* How much memory space is reserved for each of the variables declared in the above examples?

Constants

- Categorized as constant values and constant identifiers.
- *Constant Values:* _____
- *Constant Identifiers:* An identifier declared by the programmer to denote (or name) a constant value through constant declaration.

Constant identifier declaration

- Syntax:
 - **const [< type >] <idf> = < const-expression>;**
- Description:
 - The identifier (<idf>) is declared to be a constant whose value is the result of evaluating the <const-expression>.
- Note: If <type> is omitted, “integer” type is assumed.

Constant identifier declaration

- *Examples:*

- `const            x = -5;`
- `const double height = 3.667;`
- `const gender = 'M';`
- `const float pi = 3.1415926;`
- `const float dpi = 2*pi;`

Reading Assignment

Describe the benefits of using constant identifiers?

Functions

- User-defined functions
Example: `main()`
- Pre-defined functions

Predefined Functions (also called built-in functions)

- These are functions that are assumed to be common to many programmers. Predefined functions came with the compiler.
- *Examples:*
 - `pow(x, y)`: Computes x to the power of y
 - `strlen(<string>)`: Returns the length of a string
 - `clrscr( )`: clears the screen
 - `getch( )`: prompts the user to press a key

Expression

- An expression is a proper combination of constants, variables or function callings with zero or more operators.
- Examples: Assuming the following declarations,
 - `int x = -3, y;`
 - `const float pi = 3.1415926;`
- The following are valid expressions:
 - `2 + x`
 - `x * 3 + y`
 - `2 * pi < -4.5`
 - `pow (pi, x)`

Expressions

- More examples ...
 - 5
 - y
 - 'A'
 - "Enter the number of data you have > "