C++ basics (cont’d)

- Structure of a C++ program
- #include <iostream> //preprocessor
- using namespace std;
- // Comment
- /* multiple line comment
- here */
- //main functions
- int main()
- {
- int a; //variable
- cout << “My first program”;
- return 0;
- }

Variable & Data types

- **A variable** is a symbol or a name that stands for a value. e.g consider; *a + b ; a &b are variables*
- Variables enable programmers to write flexible programs (programmers write code with variables which are then replaced with real data during program execution)
- **Identifiers:** A valid identifier is a sequence of one or more letters, digits or underscore character (_).

Identifiers

- A sequence of one or more letters,etc
- No spaces , punctuation marks or symbols.
- Only letters, digits and single underscore characters are valid.
- Variable identifiers always begin with a letter.
- Can also begin with an underscore character.
- They should not match any of the C++ language keywords

Keywords

- asm, auto, bool, break, case, catch, char, class,
- const, const_cast, continue, default, delete, do,
- double, dynamic_cast, else,
- false, float, for, friend, goto
- mutable, namespace, new, operator, private,
- protected, public, register,
- short, signed, sizeof, static,
- switch, template, this, throw, true, try,
- typeid, typename, union, unsigned, using, virtual,
- void, volatile,etc

Tips

- C++ is case sensitive
- Avoid variable names that differ only in case
- Use meaningful and descriptive names
- Be consistent
  - Capitalize first letters
  - Separating words with underscore

Data types

- Variables are stored in a computers memory
- PC has to know what kind of data we want to store in the memory.
- Memory is organized in bytes.
- A byte is the minimum amount of memory that can be managed in C++
- The table below gives some data types in C++

Signed & Unsigned

- Part of the information stored about numeric data is its sign, that is + or - . This is normally stored in the sign bit(1).**By default,data is signed**
- Eg. an integer is typically allocated 2 bytes of memory. The range of possible values for an integer is -32768 to 32767. If we don't use the sign bit, the range of possible values is: 0 to 65535. When we use the **unsigned modifier, we are modifying how** the memory is used.

C++ data types

Type	Description	Size	Range
char	Character	1	Signed: -128 to 127 Unsigned: 0 to 255
short int	Short integer	2	Signed: -32768 to 32767 Unsigned: 0 to 65535
int	integer	4	Signed: -2,147,483,648 to 2,147,483,647 Unsigned: 0 to 4294967295
bool	Boolean (true or false)	1	True or false
float	Floating point number	4	3.4 e+/- 38 (7 digits)
double	Double precision floating point number	8	1.7 e+/- 308 (15 digits)
long double	Long double precision floating point number	8	1.7 e+/- 308 (15 digits)

Scope of variables

- Global variables are declared in the main body of the source code, outside all functions and can be referred to anywhere in the code
- Local variables are declared within the body of a function or a block

Initialization

```
int a=2;      //a=2
int a(2);     //a=2
int a;        //value of a is undetermined
```

Constants

- #define PI 3.142
- const datatype variable=value;

e.g const int width=20;

Operators

C++ has the following operators:

- Assignment (=) this is used to assign a value to a variable.
- Arithmetic operators

+ addition

- subtraction

* Multiplication

/ division

% module (gives the remainder of the division of 2 integers)

Operators

Compound operators (**+=, - =,^=, |=,*=, /=, %=, >>=, <<=, &=)**

e.g value += increase; is equivalent to value=value+ increase

Increase / Decrease (++/--)

– The above are equivalent to +=1 or -=1

– a++

– a+=1

– a=a+1

Increase ++ and Decrease --

- A characteristic of this operator is that it can be used both as a *prefix or as a suffix. E.g A++ or ++A may have different meanings .*
- *A++; Value stored is increased after being evaluated*
- *++A; Value is increased before being evaluated*
- *A=2;*
- *b=A++;   //A=2 b=3*
- *A=2;*
- *b=++A;   //A=3 B=3*

Relational operators

- For comparison between two expressions e.g
- == equal to
- > greater
- != not equal (different)
- < less than
- >= greater or equal to
- <= less than or equal to

Logical operators

- For logic e.g.
- ! NOT (for inversion) *!(5==5) evaluates to false*
- && AND
- || OR

Conditional operators

- It evaluates an expression and returns a different value according to the evaluated expression, depending on whether it is true or false
- Format is: *condition* ***? Result1:Result2***
- If *condition is* ***true*** the expression will return **Result1,** if not it will return ***Result2***.

Type casting

- Allows to convert a datum of a given type to another.
- Parenthesis () are used.
- E.g

```
int a;
float b;
b=(float)a; // a is now a float
b=float(a); // a is also now a float
```

Output/Input to console

- Console is a basic interface of a computer (keyboard and screen)
- cin >> a; //input a
- cout << a; //output to console –screen

Next control structures