

G205

Fundamentals of Computer Engineering

CLASS 2

Stefano Basagni

Fall 2003

M-W, 9:50am-11:30am, 410 EII

Algorithms Description

◆ Pseudo-code:

- Effective way of describing algorithms
- Highlights an algorithm characteristics and properties
- Closer to natural language
- Abstracts from unnecessary details
- It is still "code:" favors analysis and correctness proof

From Algorithms to Programs

- ◆ Pseudo-code is generally not executable by a computer
- ◆ Programs are
- ◆ Choice of the appropriate programming language:
 - Depends on particular problem
 - Depends on which language is supported

C++, 1

- ◆ Our language of choice is C++
- ◆ It has been recently standardized
- ◆ It is an example of Object Oriented Programming (OO) language
- ◆ It comes with libraries and ...
- ◆ The Standard Template Library (STL)

C++, 2

- ◆ C++ with STL = executable which are close to the pseudo code
 - Our aim: one-to-one
- ◆ Enhanced readability
- ◆ Ease correctness proof of the implementation

The C++ Environment

- ◆ Editor: Writing the code (.cpp file)
- ◆ Preprocessor: Processes the code
- ◆ Compiler: Creates the object code (OC)
- ◆ Linker: OC + libraries = executable
- ◆ Loader: Puts executable in memory
- ◆ CPU: Executes the program

A C++ Program, 1

◆ Preprocessor directives

- `#include <iostream>`

The preprocessor includes the content of the `iostream` header file (I/O)

◆ Function prototypes

- `void InsertionSort(int A[], int n);`

A C++ Program, 2

- ◆ Main function, body of the program

- `int main() {`

- `...`

- `return 0;`

- `}`

- ◆ Function definitions

- `void InsertionSort(int A[], int n) { ... }`

Example: Insertion Sort

PSEUDO CODE

```
Insertion-Sort(A, n)

  for j = 2 to n do
    key = A[j]
    i = j - 1
    while (i > 0) and (A[i] > key) do
      A[i+1] = A[i]
      i = i - 1
    A[i+1] = key
```

C++ FUNCTION

```
void InsertionSort(int A[], int n) {
  int i, key;
  for (int j=1; j<n;j++) {
    key = A[j];
    i=j-1;
    while (i >= 0) && (A[i] > key) {
      A[i+1] = A[i];
      i--; }
    A[i+1] = key;
  }
}
```

Basic C++

◆ It is like C:

- Memory concept, assignment, inc/dec
- Arithmetic, logical operators
- Equality and relational operators

◆ INPUT/OUTPUT, it is easier ...

- INPUT: `std::cin >> varName;`
- OUTPUT: `std::cout << varName;`

Control Structures

◆ Sequence Structures

- Built into C++, statements are executed one after another

◆ Selection Structures, three types:

- (1) **if** (2) **if/else** (3) **switch**

◆ Repetition Structures, three types:

- (1) **while** (2) **do/while** (3) **for**

Functions

- ◆ Pre-packaged functions:
 - C++ standard library: math, strings, characters, I/O, error checking
- ◆ User-defined functions
 - Modularize a program
 - Software reuse
 - Avoid repeated code (less errors)

Function Definition

```
#include <iostream>
using std::cout;
using std::cin;
int square(int); // function prototype
int main() {
    for (int x=1;x<11;x++)
        cout << square(x) << " "; // function call
    cout << endl;
    return 0;
}
int square(int y) { // function definition
    return y*y;
}
```

Assignments

- ◆ Deitel & Deitel book, first five chapters
- ◆ Updated information on the class web page:

www.ece.neu.edu/courses/eceg205/2003fa