Scientific Programming

Xiaoqiang Wang
Florida State University

Tentative Class Schedule

1. A quick start to C programming
2. Introduction
3. Compilation, linking, making and debuging
4. Object Oriented Programming and C++ I
5. C++ pointers and references
6. Object Oriented Programming and C++ II
7. Code Structure and Code Documentation – DOxygen

Tentative Class Schedule

8. Data Structure
9. Standard Template Library (STL)
10. Parallel Programming
11. Project: streamtube generation (C++)
12. Python: Introductory

Class Expectations

- Write many programs
- Take notes in class (IMPORTANT)
- Ask questions (IMPORTANT)
- Modify examples (what-if scenarios)
- Experimentation with code
- Benchmark and document all codes; output the benchmark results
- Write programs clearly (affects the grade)

Syllabus Highlights

- Class attendance is mandatory
- No Final project or exam
- Weekly homeworks
  - Articles to read and summarize
  - Programming assignments
- Occasional quizzes
- Take notes in class
- Ask questions

What we will do

- Implement some basic algorithms
- Learn to translate algorithms into “pseudocode”
- Learn to translate pseudo-code to code in a top-down approach
- Learn to think in an object-oriented way
- Learn the major features of each language
- Use the Web to lookup information dynamically
- Program in class in real time to demonstrate concepts

What we will not do

- Learn theoretical properties of algorithms and data structures
- Learn every last feature of each language

The Language List: C

- C: 1972
  - C was originally a systems language for Unix on the PDP-11, briefly named NB. It was influenced by BCPL through Thompson’s B. C is terse, low-level and permissive. Preprocessor. Partly due to its distribution with Unix, C became the language most widely used for software implementation.
- ANSI C: 1989
  - A revision of C, adding function prototypes, structure passing, and assignment, and standardized library functions. ANSI X3.159-1989.
- GNU C: 1992
  - Many extensions: compound statement within an expression, pointers to labels, local labels, nested functions, typeof operator, compound and conditional expressions and casts allowed as lvalues, long long ints, arrays of variables length macros with variable number of arguments, non-constant initializers, constructor expressions, labeled elements in initializers, case ranges, variable attributes.

The Language List: C++

- C++: 1986
  - An object-oriented superset of C. In C++ a class is a userdefined type, syntactically a struct with member functions. Constructors and destructors are member functions called to create or destroy instances. A friend is a non-member function that is allowed to access the private portion of a class. C++ allows implicit type conversion, function inlining, overloading of operators and function names, default function arguments, and pass by reference. It has streams for I/O.
- C++ release 3.0: 1996
  - Templates added

Generic Structure of a Scientific Code

- Data Input
- Initializations
- Computations
- Data Output

Important for Scientific Coding

- Minimize hard coding (numbers and strings)
- Documentation
- Good Mnemonics
- Restart capability
- Documented data output
- Code testing
- Data analysis and visualization
- Benchmarking
- Profiling

Language Concepts

- Variables (x,y,_a45,..)
- Types (int, real, float, struct, typedef,..)
- Classes, structures, modules
- Subclasses, inheritance, interfaces
- Methods, subroutines, functions
- Function and operator overloading
- Pointers, references, values

Language Commonalities

- Variables
- Object-orientation (encapsulation)
- Subroutines/functions/methods
- Iterators (for, while, do …)
- Flow control (if, unless, …)
- Multi-file objects
- Input/Output (I/O)
- External libraries
- Project-related tools (make, ant, maven)

C++

- Object-oriented (but objects are not necessary)
- Superset of C
- Low level, close to hardware devices
- Multiple inheritance (dangerous)
- Function and operator overloading
- Pointers and references
- Exception handling
- Easy to write code difficult to debug
- Hard to construct generic tools
  - Due to aliasing and pointer manipulation

C++: some disadvantages

- Messy class operations
  - Friends, multiple inheritance
- Include files can be dangerous
- No exponentiation operator
- Use of pointers complicates debugging
- Arguments can be passed by reference or by pointer (can be confusing)
- Many others, … but

THE LANGUAGE IS VERY POWERFULL!!

Popularity Ranking of Languages

- Google “TIOBE Index”