C++ for Non C Programmers

Topics

Getting Started

- First program (hello.cpp)
- Compile steps
- How to compile and run a program

Data Types and Variables

- Fundamental data types
- Data type values and sizes
- Variable declarations
- Variable names
- Constants
- Character constants
- String constants

Operators and Expressions

- What are expressions?
- Arithmetic operators
- Relational operators
- Assignment operator
- Expressions have resulting values
- True and False
- Logical operators
- Increment and decrement operators (++ and --)
- 'Operate-Assign' operators (+=, *=, ...)
- Conditional expression
- Operator precedence
- Precedence and order of evaluation
- Evaluation of logical operators
- Type conversions
- Cast operator

Control Flow

- Statements
- if - else
- if() - else if ()
- switch()
- while()
- do - while()
- for()
- The for() loop - diagram
- Break statement
- Continue statement

Functions

- What is a function?
- Why use functions?
- Anatomy of a function
- Arguments passed by value
- When to use the return Statement
- Returning non-integer values
- Functions in multiple source files
- Concept of variable scope
- Automatic variables
- Global (external) variables
- Static variables
- External static variables

Pointers and Arrays

- What is a pointer?
- Pointer operators
- Why use pointers?
- Arrays (a picture)
- & operator
- Pointers and arrays

- Passing arrays to functions
- Initializing arrays
- Strings and character pointers
- What is char s[7] ???
- Arrays of pointers - diagram
- Command-line arguments

Structures

- Comparison of structures and arrays
- Structure definitions and declarations
- Arrays of structures

Classes

- Creating a data structure
- Methods
- Object scope
- C++ input and output
- Namespaces
- Data abstraction
- Enforcing data encapsulation
- File organization
- Classes in C++
- Objects
- this Pointer

Constructors and Destructors

- Debug output
- Default constructor
- When are constructors called?
- Destructor
- Copy constructor
- Other Constructors
- Why did it work before?
- Composition

- Report class
- Code reuse
- Initialization lists

Inheritance

- Inheritance
- Bugreport
- Protected access modifier
- Access and inheritance
- Constructors and inheritance
- Initialization lists revisited
- Multiple inheritance

Virtual Functions

- Inheritance and assignment
- Inside Report's assignment operator
- Using pointers - a quick look at basics
- Class assignment and pointers
- Static binding
- Dynamic binding
- Polymorphism
- Using the show_rep() function
- Designing member function inheritance

Pure Virtual Functions

- Bugfix and its relationship with Bugreport
- Using Bugfix with show_rep()
- Adding Bugfix to the hierarchy
- Coding for the document class
- Reexamining the document class
- Pure virtual functions
- Updated: Designing member function inheritance

References and Constants

- References

- Displaying references
- Changing references
- Pass by reference
- Returning by reference
- Constant variables
- Constant references
- Constant methods

New and delete

- new and delete
- Array allocation
- Report class
- Compiler version of the copy constructor
- Guidelines for copy constructors
- Report constructors and new
- Report destructor and delete
- Virtual destructors

Casting in C++

- Casting: a review
- New casting syntax
- Creating a string class
- String class
- Conversion constructor
- Expanding our casting options
- Using the Casting Operator

Class Methods and Data

- Class data
- Class methods
- Using the new data
- More on class methods

Overloaded Functions

- Function overloading

- Using overloaded functions
- Rules for overloading
- Overloading based on const-ness
- Default arguments
- Invoking functions with default arguments

Overloaded Operators

- Basics of overloading
- Overloading operator+
- Coping with commutativity
- Non-Commutative operators
- friends and their problems
- Assignment operator
- Overloading the << Operator
- Using date with cout

Exception Handling

- Why exception handling?
- try/catch/throw
- Exception classes
- Standard exception hierarchy
- Multiple catch blocks
- Catching everything
- Unhandled exceptions
- Exception in constructors and destructors
- Designing for exceptions

Standard Template Library

- Class template concepts
- Standard Template Library (STL) overview
- Containers
- Iterators
- Iterator syntax
- Non-Mutating sequential algorithms

- Mutating sequential algorithms
- Sorting algorithms
- Numeric algorithms
- auto_ptr class
- string class

STL Containers

- Container classes
- Container class algorithms
- vector class
- Additional vector class methods
- deque class
- list class
- set and multiset classes
- map and multimap classes

Appendix A: Reference Sheets

- Constants, references and pointers
- Input/Output
- this Pointer
- Complete Report/Document Hierarchy

Appendix B: Templates

- Scenario
- Designing an array class
- Code for FloatArray
- Code for IntArray
- Templates
- Template syntax
- Using templates
- Using classes with templates
- Additional template features
- Standard Template Library

Appendix C: Sample Problems