

Display a proposed outline for the next semester

Changed sections have a light blue background

Subject code

BTP300

Subject title (name)

Object-oriented Software Development I - C++

Date and time of most recent update on SSOS

2013-06-23 15:08:44 for Fall - 2013

Description

This subject expands the student's skill-set in object-oriented programming and introduces the student to threaded programming. The student learns to model relationships between classes using containers, inheritance hierarchies and polymorphism in the C++ programming language and to write C++ programs that execute on multiple threads.

Mode of instruction

2 hours interactive lecture per week.
2 hours activity-based learning per week.

Learning outcomes

- design collections of model objects using sequential containers and multi-dimensional arrays to solve a complex systems or business problem
- create function objects and closures to customize a programming solution for a particular application
- model generalization and specialization using single and multiple inheritance to minimize the duplication of code in complex hierarchies
- model polymorphic behavior using interfaces, virtual functions and templates (generics) to amplify the reusability of code
- implement design components using algorithms of the standard template library to utilize existing

technologies

- create program components of quadratic complexity to solve non-linear problems
- design program components using raw pointers, pointer arithmetic and smart pointers to access data in program memory
- design multi-tasked solutions using threading libraries to improve the performance of a program
- design file stream objects to backup text and binary data for future restoration
- trace the execution of program code that includes a linked list to debug an application

Topic outline

- Introduction – 8%
 - variable and object scope
 - program, file, class, function and block scope
 - namespaces (review, anonymous, inline)
 - stages of compilation
 - pre-processing
 - macros, conditionals, pragmas
 - compilation
 - static assertions
 - linkage
 - external, internal, no linkage
 - options (g++, Visual Studio)
- Fundamental Types – 8%
 - scalar types
 - integral representation
 - floating-point representation (IEEE 754)
 - size specifiers
 - range specifiers (signed, unsigned)
 - unicode, wide characters
 - pointer types
 - nullptr
 - generic
 - enumerations
 - synonyms
 - auto
 - storage duration and linkage (extern, static)
- Compound Types - 8%
 - arrays
 - multi-dimensional
 - dynamic allocation
 - ragged
 - representation of matrices and vectors
 - classes
 - initializer lists
 - class variables, class methods
- Class Relationships – 20%
 - inheritance
 - generalization and specialization

- review of abstract base classes and interfaces
 - virtual functions
 - multiple inheritance
 - virtual inheritance
 - polymorphism
 - review of polymorphism and type systems
 - class templates and specialization
 - composition
 - container classes
 - flexibility of low coupling
- Logic - 12%
 - expressions
 - lvalues, rvalues, constant operands
 - constexpr
 - post-fix, pre-fix, unary, binary, ternary
 - bit-wise operations
 - constrained casts
 - range-based for
 - functions
 - linkage
 - function pointers
 - function objects
 - closures (lambda expressions)
 - optional return type syntax
 - decltype
 - error handling
 - command line input
 - return codes to operating systems
 - exceptions (throw, try, catch)
 - exit
- Standard Template Library – 12%
 - components (containers, function objects, iterators)
 - conceptual description of linked lists
 - tracing linked list logic
 - sequential containers
 - array
 - vector
 - deque
 - applications
 - custom container classes
 - algorithms (find, sort, copy)
 - iterators
- Performance - 16%
 - core language facilities
 - rvalue references and move semantics
 - alignment control
 - multi-threading
 - introduction to concurrent programming
 - pthreads library (mutex, condition)

- OpenMP library (parallel for, critical)
- Memory Model - 8%
 - review of raw pointers and references
 - pointer arithmetic
 - smart pointers
- Persistence - 5%
 - file stream objects
 - text access
 - binary access
- Related Topics - 3%
 - C11 language
 - structs, unions
 - anonymous
 - bit-fields
 - linking multi-language binaries
 - break, continue, goto
 - comparison of C++ and C syntax
 - C++14 and C++17

Prescribed text(s)

- Intermediate C++11 - June 2014 Edition by Chris Szalwinski (available at the Seneca bookstore)

Reference material

- The C++ Programming Language, 3rd Edition or Special Edition; Bjarne Stroustrup; Addison Wesley; ISBN 0-201-88954-4 (3rd Ed.) or 0-201-70073-5 (Special Ed.)

Supplies

- None

Evaluation

- Assignments (minimum 3) - 30%
- Workshops (minimum 10) - 20%
- Test and Quizzes - 20%
- Final Exam - 30%

Editor's comments (Chris)

(none)

Reviewer's comments (peter.mcintyre)

(none)

Publisher's comments (daman.panesar)

(none)