

Space Visions Congress

Seminar: C++ Programming Language

When: Friday April 27, 2007 8:00 a.m. to 12:00 a.m.

Where: University of Central Florida Solar Energy Center (FSEC), 1519
Clearlake Road, Cocoa, Florida

Instructor: Ali Shaykhian

Space Visions Congress is sponsored by the Canaveral Council of Technical Societies (CCTS). CCTS is a voluntary, not-for-profit association of engineering, technical, and scientific societies that support memberships who live and work along Florida's Space Coast.

C++ Programming Language: The C++ seminar covers the fundamentals of C++ programming language. The C++ fundamentals are grouped into three parts where each part includes both concept and programming examples aimed at for hands-on practice. The first part covers the functional aspect of C++ programming language with emphasis on function parameters and efficient memory utilization. The second part covers the essential framework of C++ programming language, the object-oriented aspects. Information necessary to evaluate various features of object-oriented programming; including encapsulation, polymorphism and inheritance will be discussed. The last part of the seminar covers template and generic programming. Examples include both user defined and standard templates.

Specific expectations are:

- Create, compile and run C++ programs
- Read, recognize, and describe C++ syntax
- Recognize functions, decisions, loops and exceptions
- Differentiate among value, reference and pointer parameters
- Declare, define and use variables, constants, arrays, pointers and references
- Compare and contrast data storage, stack storage, and heap storage
- Define and implement classes to represent real objects
- Learn to implement object-oriented designs, emphasize on encapsulation, inheritance and polymorphism
- Demonstrate the benefits of user define function templates and class templates
- Show the use of standard template libraries

Detail Outline

First C++ program, using the editor. Compile, link and run a program

Basic program construction

Functions, program statements

Preprocessor directives

Using blocks of code, semicolons and positioning, indentation practices

Data, declarations and definitions

Operators, arithmetic, increment, relational, conditional, and logical operators

Program control statements, the if statement

The conditional expression, the switch statement

The for loop, the while loop, the do-while loop
Functions, the function declaration, calling the function, the function definition
Passing arguments to functions, passing by value, by reference, by pointer
Overloaded functions, different numbers of arguments, default arguments
Inline functions
Storage classes, automatic variables, external variables, static variables
Structures, defining a structure variable, accessing structure members
The pointer operators
Assigning values through a pointer, pointer expressions
Pointer comparisons, pointers and arrays
Procedural languages, the object-oriented approach
Characteristics of object-oriented languages
Classes and objects, specifying the class, the this pointer
Constructors, destructors
Overloaded constructors
Copy operations, copy constructor and operator=
Static class data and function
Inline functions, friend functions, static functions
Arrays of objects
Pointers to objects
Assigning objects, creating and using a copy constructor
Operator overloading, operator overloading using member functions
Using member functions to overload unary operators
Inheritance "Is-A", derived class and base class
Virtual functions, and polymorphism
Member access and inheritance, base class access control
Using protected members
Constructors and inheritance, calling base class constructors
Creating a multilevel hierarchy, inheriting multiple base classes
When constructor and destructor functions are executed
Pointers to derived types, references to derived types
Virtual functions and polymorphism
Applying virtual functions
Pure virtual functions and abstract classes
Function and class template
Standard template libraries