TOPICS COVERED IN FIRST HALF OF COURSE

struct. Concept of, syntax of defining, dot operator for referencing members.

Class. How different from struct. Concept of "object" with both data and methods (functions). Concept of public and private members. "Encapsulation" of data away from direct access by caller.

Concept of "accessor" and "mutator" as means to work with private data members.

Syntax of declaring a "member" function. Use of and need for the scope-resolution operator ::

Constructor. WHY "constructor", rather than "initializer"? Default constructor, constructors with arguments, "overload" of constructor function name, purpose of constructors. Different ways (syntaxes) in which values can be provided to constructor (e.g., as defaults assigned to variables within the class declaration, as assignment statements within the body of the constructor, as initializers on the same line as the function header). WHEN constructors are invoked (e.g., automatically upon declaration, or by user when calling constructor as function).

Concept of "Abstract Data Type" ADT. Notion that you should be able to change the underlying implementation of things within an ADT, with NO effect upon how the caller interacts with objects.

Friend function. WHY do they exist? What is the major advantage of having one? When is it appropriate to write one? Syntax of how to define, declare, use.

Overloaded operator. Syntax of different kinds (unary, binary) – how to declare and define and use.

"const" designator. When to use on function argument, when to use on entire function.

Overloading the << and >> operators. Need for function to return a "stream" value. Syntax of how to define, declare, use.

Placement of class "objects" within an array. Syntax of referencing member variables and data.

Concept of an array's name being fundamentally equal to a pointer to the array's 0^{th} element.

The -> operator, syntax of using it to reference members of object being pointed to.

Placement of an array WITHIN a class "object". Syntax of use and referencing.

Issues with use of dynamic array within "object". Necessity to handle the dynamic data correctly.

Destructor. WHY needed? When invoked? Major purpose of.

Copy constructor and overloaded "=" assignment operator. WHY needed? What should they accomplish? WHEN invoked?

Interaction between "scope" of an object and timing of invocation of constructors/destructor.

Memory model of how dynamic variables differ from static ones

Concepts of LIFO (i.e., like a "stack") and FIFO access to a sequential container.