Appendix E

Glossary of Object Oriented Terms

abstract class: A class primarily intended to define an instance, but can not be instantiated without additional methods.

abstract data type: An abstraction that describes a set of items in terms of a hidden data structure and operations on that structure.

abstraction: A mental facility that permits one to view problems with varying degrees of detail depending on the current context of the problem.

accessor: A public member subprogram that provides query access to a private data member.

actor: An object that initiates behavior in other objects, but cannot be acted upon itself.

agent: An object that can both initiate behavior in other objects, as well as be operated upon by other objects.

ADT: Abstract data type.

AKO: A Kind Of. The inheritance relationship between classes and their superclasses.

allocatable array: A named array having the ability to dynamically obtain memory. Only when space has been allocated for it does it have a shape and may it be referenced or defined.

argument: A value, variable, or expression that provides input to a subprogram.

array: An ordered collection that is indexed.

array constructor: A means of creating a part of an array by a single statement.

array overflow: An attempt to access an array element with a subscript outside the array size bounds.

array pointer: A pointer whose target is an array, or an array section.

array section: A subobject that is an array and is not a defined type component.

assertion: A programming means to cope with errors and exceptions.

assignment operator: The equal symbol, "=", which may be overloaded by a user.

assignment statement: A statement of the form "variable = expression".

association: Host association, name association, pointer association, or storage association.

attribute: A property of a variable that may be specified in a type declaration statement.

automatic array: An explicit-shape array in a procedure, which is not a dummy argument, some or all of whose bounds are provided when the procedure is invoked.

base class: A previously defined class whose public members can be inherited by another class. (Also called a super class.)

behavior sharing: A form of polymorphism, when multiple entities have the same generic interface. This is achieved by inheritance or operator overloading.

binary operator: An operator that takes two operands.

bintree: A tree structure where each node has two child nodes.

browser: A tool to find all occurrences of a variable, object, or component in a source code.

call-by-reference: A language mechanism that supplies an argument to a procedure by passing the address of the argument rather than its value. If it is modified, the new value will also take effect outside of the procedure.

call-by-value: A language mechanism that supplies an argument to a procedure by passing a copy of its data value. If it is modified, the new value will not take effect outside of the procedure that modifies it.

class: An abstraction of an object that specifies the static and behavioral characteristics of it, including their public and private nature. A class is an ADT with a constructor template from which object instances are created.

class attribute: An attribute whose value is common to a class of objects rather than a value peculiar to each instance of the class.

class descriptor: An object representing a class, containing a list of its attributes and methods as well as the values of any class attributes.

class diagram: A diagram depicting classes, their internal structure and operations, and the fixed relationships between them.

class inheritance: Defining a new derived class in terms of one or more base classes.

client: A software component that users services from another supplier class.

concrete class: A class having no abstract operations and can be instantiated.

compiler: Software that translates a high-level language into machine language.

component: A data member of a defined type within a class declaration

constructor: An operation, by a class member function, that initializes a newly created instance of a class. (See default and intrinsic constructor.)

constructor operations: Methods which create and initialize the state of an object.

container class: A class whose instances are container objects. Examples include sets, arrays, and stacks.

container object: An object that stores a collection of other objects and provides operations to access or iterate over them.

control variable: The variable which controls the number of loop executions.

data abstraction: The ability to create new data types, together with associated operators, and to hide the internal structure and operations from the user, thus allowing the new data type to be used in a fashion analogous to intrinsic data types.

data hiding: The concept that some variables and/or operations in a module may not be accessible to a user of that module; a key element of data abstraction.

data member: A public data attribute, or instance variable, in a class declaration.

data type: A named category of data that is characterized by a set of values. together with a way to denote these values and a collection of operations that interpret and manipulate the values. For an intrinsic type, the set of data values depends on the values of the type parameters.

deallocation statement: A statement which releases dynamic memory that has been previously allocated to an allocatable array or a pointer.

debugger software: A program that allows one to execute a program in segments up to selected breakpoints, and to observe the program variables.

debugging: The process of detecting, locating, and correcting errors in software.

declaration statement: A statement which specifies the type and, optionally, attributes of one or more variables or constants.

default constructor: A class member function with no arguments that assigns default initial values to all data members in a newly created instance of a class.

defined operator: An operator that is not an intrinsic operator and is defined by a subprogram that is associated with a generic identifier.

deque: A container that supports inserts or removals from either end of a queue.

dereferencing: The interpretation of a pointer as the target to which it is pointing.

derived attribute: An attribute that is determined from other attributes.

derived class: A class whose declaration indicates that it is to inherit the public members of a previously defined base class.

derived type: A user defined data type with components, each of which is either of intrinsic type or of another derived type.

destructor: An operation that cleans up an existing instance of a class that is no longer needed.

destructor operations: Methods which destroy objects and reclaim their dynamic memory.

domain: The set over which a function or relation is defined.

dummy argument: An argument in a procedure definition which will be associated with the actual (reference or value) argument when the procedure is invoked.

dummy array: A dummy argument that is an array.

dummy pointer: A dummy argument that is a pointer.

dummy procedure: A dummy argument that is specified or referenced as a procedure.

dynamic binding: The allocation of storage at run time rather than compile time, or the run time association of an object and one of its generic operations..

edit descriptor: An item in an input/output format which specifies the conversion between internal and external forms.

encapsulation: A modeling and implementation technique (information hiding) that separates the external aspects of an object from the internal, implementation details of the object.

exception: An unexpected error condition causing an interruption to the normal flow of program control.

explicit interface: For a procedure referenced in a scoping unit, the property of being an internal procedure, a module procedure, an external procedure that has an interface (prototype) block, a recursive procedure reference in its own scoping unit, or a dummy procedure that has an interface block.

explicit shape array: A named array that is declared with explicit bounds.

external file: A sequence of records that exists in a medium external to the program.

external procedure: A procedure that is defined by an external subprogram.

FIFO: First in, first out storage; a queue.

friend: A method, in C++, which is allowed privileged access to the private implementation of another object.

function body: A block of statements that manipulate parameters to accomplish the subprogram's purpose.

function definition: Program unit that associates with a subprogram name a return type, a list of arguments, and a sequence of statements that manipulate the arguments to accomplish the subprogram's purpose

function header: A line of code at the beginning of a function definition; includes the argument list, and the function return variable name.

generic function: A function which can be called with different types of arguments.

generic identifier: A lexical token that appears in an INTERFACE statement and is associated with all the procedures in the interface block.

generic interface block: A form of interface block which is used to define a generic name for a set of procedures.

generic name: A name used to identify two or more procedures, the required one being determined by the types of the non-optional arguments in the procedure invocation.

generic operator: An operator which can be invoked with different types of operands.

Has-A: A relationship in which the derived class has a property of the base class.

hashing technique: A technique used to create a hash table, in which the array element where an item is to be stored is determined by converting some item feature into an integer in the range of the size of the table.

heap: A region of memory used for data structures dynamically allocated and deallocated by a program.

host: The program unit containing a lower (hosted) internal procedure.

host association: Data, and variables automatically available to an internal procedure from its host.

information hiding: The principle that the state and implementation of an object should be private to that object and only accessible via its public interface.

inheritance: The relationship between classes whereby one class inherits part or all of the public description of another base class, and instances inherit all the properties and methods of the classes which they contain.

instance: A individual example of a class invoked via a class constructor.

instance diagram: A drawing showing the instance connection between two objects along with the number or range of mapping that may occur.

instantiation: The process of creating (giving a value to) instances from classes.

intent: An attribute of a dummy argument that which indicates whether it may be used to transfer data into the procedure, out of the procedure, or both.

interaction diagram: A diagram that shows the flow of requests, or messages between objects.

interface: The set of all signatures (public methods) defined for an object.

internal file: A character string that is used to transfer and/or convert data from one internal storage mode to a different internal storage mode.

internal procedure: A procedure contained within another program unit, or class, and which can only be invoked from within that program unit, or class.

internal subprogram: A subprogram contained in a main program or another subprogram.

intrinsic constructor: A class member function with the same name as the class which receives initial values of all the data members as arguments.

Is-A: A relationship in which the derived class is a variation of the base class.

iterator: A method that permits all parts of a data structure to be visited.

keyword: A programming language word already defined and reserved for a single special purpose.

LIFO: Last in, first out storage; a stack.

link: The process of combining compiled program units to form an executable program.

linked list: A data structure in which each element identifies its predecessor and/or successor by some form of pointer.

linker: Software that combines object files to create an executable machine language program.

list: An ordered collection that is not indexed.

map: An indexed collection that may be ordered.

matrix: A rank-two array.

member data: Variables declared as components of a defined type and encapsulated in a class.

member function: Subprograms encapsulated as members of a class.

method: A class member function encapsulated with its class data members.

method resolution: The process of matching a generic operation on an object to the unique method appropriate to the object's class.

message: A request, from another object, for an object to carry out one of its operations.

message passing: The philosophy that objects only interact by sending messages to each other that request some operations to be performed.

module: A program unit which allows other program units to access variables, derived type definitions, classes and procedures declared within it by USE association.

module procedure: A procedure which is contained within a module, and usually used to define generic interfaces, and/or to overload or define operators.

nested: Placement of a control structure inside another control structure.

object: A concept, or thing with crisp boundaries and meanings for the problem at hand; an instance of a class.

object diagram: A graphical representation of an object model showing relationships, attributes, and operations.

object-oriented (OO): A software development strategy that organizes software as a collection of objects that contain both data structure and behavior. (Abbreviated OO.)

object-oriented programming (OOP): Object-oriented programs are object-based, class-based, support inheritance between classes and base classes and allow objects to send and receive messages.

object-oriented programming language: A language that supports objects (encapsulating identity, data, and operations), method resolution, and inheritance.

octree: A tree structure where each node has eight child nodes.

OO (acronym): Object-oriented.

operand: An expression or variable that precedes or succeeds an operator.

operation: Manipulation of an object's data by its member function when it receives a request.

operator overloading: A special case of polymorphism; attaching more than one meaning to the same operator symbol. 'Overloading' is also sometimes used to indicate using the same name for different objects.

overflow: An error condition arising from an attempt to store a number which is too large for the storage location specified; typically caused by an attempt to divide by zero.

overloading: Using the same name for multiple functions or operators in a single scope.

overriding: The ability to change the definition of an inherited method or attribute in a subclass.

parameterized classes: A template for creating real classes that may differ in well-defined ways as specified by parameters at the time of creation. The parameters are often data types or classes, but may include other attributes, such as the size of a collection. (Also called generic classes.)

pass-by-reference: Method of passing an argument that permits the function to refer to the memory holding the original copy of the argument

pass-by-value: Method of passing an argument that evaluates the argument and stores this value in the corresponding formal argument, so the function has its own copy of the argument value

pointer: A single data object which stands for another (a "target"), which may be a compound object such as an array, or defined type.

pointer array: An array which is declared with the pointer attribute. Its shape and size may not be determined until they are created for the array by means of a memory allocation statement.

pointer assignment statement: A statement of the form "pointer-name $\Rightarrow$ target".

polymorphism: The ability of an function/operator, with one name, to refer to arguments, or return types, of different classes at run time.

post-condition: Specifies what must be true after the execution of an operation.

pre-condition: Specifies the condition(s) that must be true before an operation can be executed.

private: That part of an class, methods or attributes, which may not be accessed by other classes, only by instances of that class.

protected: (Referring to an attribute or operation of a class in C++) accessible by methods of any descendent of the current class.

prototype: A statement declaring a function's return type, name, and list of argument types.

pseudocode: A language of structured English statements used in designing a step-by-step approach to solving a problem.

public: That part of an object, methods or attributes, which may be accessed by other objects, and thus constitutes its interface.

quadtree: A tree structure where each tree node has four child nodes.

query operation: An operation that returns a value without modifying any objects.

rank: Number of subscripted variables an array has. A scalar has rank zero, a vector has rank one, a matrix has rank two.

scope: That part of an executable program within which a lexical token (name) has a single interpretation.

section: Part of an array.

sequential: A kind of file in which each record is written (read) after the previously written (read) record.

server: An object that can only be operated upon by other objects.

service: A class member function encapsulated with its class data members.

shape: The rank of an array and the extent of each of its subscripts. Often stored in a rank-one array.

side effect: A change in a variable's value as a result of using it as an operand, or argument.

signature: The combination of a subprogram's (operator's) name and its argument (operand) types. Does not include function result types.

size: The total number of elements in an array.

stack: Region of memory used for allocation of function data areas; allocation of variables on the stack occurs automatically when a block is entered, and deallocation occurs when the block is exited

stride: The increment used in a subscript triplet.

strong typing: The property of a programming language such that the type of each variable must be declared.

structure component: The part of a data object of derived type corresponding to a component of its type.

sub-object: A portion of a data object that may be referenced or defined independently of other portions. It may be an array element, an array section, a structure component, or a substring.

subprogram: A function or subroutine subprogram.

subprogram header: A block of code at the beginning of a subprogram definition; includes the name, and the argument list, if any.

subscript triplet: A method of specifying an array section by means of the initial and final subscript integer values and an optional stride (or increment).

super class: A class from which another class inherits. (See base class.)

supplier: Software component that implements a new class with services to be used by a client software component.

target: The data object pointed to by a pointer, or reference variable.

template: An abstract recipe with parameters for producing concrete code for class definitions or subprogram definitions.

thread: The basic entity to which the operating system allocates CPU time.

tree: A form of linked list in which each node points to at least two other nodes, thus defining a dynamic data structure.

unary operator: An operator which has only one operand.

undefined: A data object which does not have a defined value.

underflow: An error condition where a number is too close to zero to be distinguished from zero in the floating-point representation being used.

utility function: A private subprogram that can only be used within its defining class.

vector: A rank-one array. An array with one subscript.

vector subscript: A method of specifying an array section by means of a vector containing the subscripts of the elements of the parent array that are to constitute the array section.

virtual function: A genetic function, with a specific return type, extended later for each new argument type.

void subprogram: A C++ subprogram with an empty argument list and/or a subroutine with no returned argument.

work array: A temporary array used for the storage of intermediate results during processing.