

CS304 - Object Oriented Programming Glossary By www.virtualians.pk

Abstract class : A class that can only be used as a base class for some other class. A class is abstract if it has at least one pure virtual function.

Access control : A C++ mechanism for prohibiting or granting access to individual members of a class. See public, private, protected, and visibility.

Access declaration : A way of controlling access to a specified member of a base class when it is used in a derived class.

Access specifier : A way of labelling members of a class to specify what access is permitted i.e public, private, and protected.

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

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

Allocation : The process of giving memory space to an object. See dynamic storage,static storage, and deallocation.

ANSI : Acronym for American National Standards Institute, a standards body currently standardizing C++.

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.

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

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.

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 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.

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.

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

constructor: : An operation, by a class member function, that initializes a newly created instance of a class.

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.

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.

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.

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

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.

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.

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

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

member function : Subprograms encapsulated as members of a class.

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

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

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

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.

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.

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.

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.)

pointer: : A single data object which stands such as an array, or defined type.

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

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.

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

super class: : A class from which another class inherits.