Object oriented definitions

abstraction is a simplified view of an object in the user's own vocabulary

Classes
A class is a programming construct that defines the common state and behavior of a group of similar objects

Objects
An object is a software entity that combines state and behavior

Method
An object's abilities. In language, methods (sometimes referred to as "functions") are verbs.Lassie, being a Dog, has the ability to bark. So bark() is one of Lassie's methods. She may have other methods as well, for example sit() or eat() or walk() or save_timmy(). Within the program, using a method usually affects only one particular object; all Dogs can bark, but you need only one particular dog to do the barking.

Message passing (interfacing) :
"The process by which an object sends data to another object or asks the other object to invoke a method.

constructor
(sometimes shortened to **ctor**) in a class is a special type of subroutine called at thecreation of an object. It prepares the new object for use, often accepting parameters which the constructor uses to set any member variables required when the object is first created. It is called a constructor because it constructs the values of data members of the class

A copy constructor
is a special constructor in the C++ programming language for creating a new object as a copy of an existing object.

object composition
is a way to combine simple objects or data types into more complex ones.(Ex: engine + doors + windows + body = Car)

dynamic binding

destructor
(sometimes shortened to **dtor**) is a method which is automatically invoked when the object is destroyed.

default constructor refers to a constructor that is automatically generated in the absence of explicit constructors

Exception handling
is the process of responding to the occurrence, during computation, of *exceptions* – anomalous or exceptional situations requiring special processing – often changing the normal flow of program execution.

Encapsulation
is A language mechanism for restricting access to some of the object's components.

Encapsulation (description)
Encapsulation conceals the functional details of a class from objects that send messages to it.
For example, the Dog *class has a* bark() *method. The code for the* bark() *method defines exactly how a bark happens (e.g., by* inhale() *and then* exhale()*, at a particular pitch and volume). Timmy,*Lassie*'s friend, however, does not need to know exactly how she barks. Encapsulation is achieved by specifying which classes may use the members of an object. The result is that each object exposes to any class a certain interface - those members accessible to that class. The reason for encapsulation is to prevent clients of an interface from depending on those parts of the implementation that are likely to change in future, thereby allowing those changes to be made more easily, that is, without changes to clients. For example, an interface can ensure that puppies can only be added to an object of the class* Dog *by code in that class. Members are often specified as* ***public****,* ***protected*** *or* ***private****, determining whether they are available to all classes, sub-classes or only the defining class.*

inheritance
is a way to reuse code of existing objects, or to establish a subtype from an existing object, or both, depending upon programming language support. In *classical inheritance* where objects are defined by classes, classes can inherit attributes and behavior from pre-existing classes called base classes, The resulting classes are known as *derived classes*

Multiple inheritance
is a feature of some object-oriented computer programming languages in which a class can inherit characteristics and features from more than one superclass. It is distinct to single inheritance, where a class may only inherit from one particular superclass.

polymorphism
is a programming language feature that allows values of different data types to be handled using a the same interface. The concept of parametric polymorphism applies to both data types and functions.

polymorphic function. A function that can evaluate to or be applied to values of different types is known as a polymorphic function.

a **virtual function** or **virtual method** is a function or method whose behavior can be overridden within an inheriting class by a function with the same signature. This concept is a very important part of the polymorphism

Inline Methods

this pointer points to the object itself;

Is-a Relationship
is shorthand for saying that an object of a derived class is substitutable for object of the base class

extensibility:
the ability to add new features to an application without significant impact to existing code.

Abstract base class :

big three : Destructor, copy constructor, and assignment operator.

Remote ownership is the responsibility that comes with being the owner of something allocated from the heap.

Operator overloading allows C/C++ operators to have user-defined meanings on user-defined types (classes).

---------- Standard template library ----------

STL a collection of templates representing containers, iterators, function objects, and algorithms

function template is a generic function description; that is, it defines a function in terms of a generic type for which a specific type, such as int or double, can be substituted.

Function objects are objects that act like functions;

vector holds a set of like values that can be accessed randomly

a container
is a class, a data structure or an abstract data type (ADT) whose instances are collections of other objects. In other words; they are used for storing objects in an organized way following specific access rules. The size of the container depends on the number of the objects (elements) it contains.

iterator is an object that enables a programmer to move through a container.

Sequence container: The sequence container classes store their objects in a linear sequence.

Associative container : The associative container classes store their objects so that the keys are in a sorted order based on a comparison object or function.

exception class defined in C++ and it has been used as a base class for other exception classes.
