typedef type newname; --🡪 typedef int feet;

feet distance;

enum color{ red, green, blue} c;

c = blue; // will contain value 2.

#include <cmath> 🡪 abs(int), sqrt(double), pow(double,double)

Derived class inherits all member functions of the base class except constructors, copy constructors, destructors, overloaded operators and friend functions of the base class.

Friend functions are declared in a class and defined outside the class. They have access to all the private members of a given class.

Singleton class – All the member functions and variables of this class are static except for the constructor and destructor

Public Inheritance: When deriving a class from a public base class, public members of the base class become public members of the derived class and protected members of the base class become protected members of the derived class. A base class's private members are never accessible directly from a derived class, but can be accessed through calls to the public and protected members of the base class.

Protected Inheritance: When deriving from C++ a protected base class, public and protected members of become protected members of the derived class.

Private Inheritance: When deriving class, public and protected members of become private members of the derived class.

Box operator+(const Box& b){}

Box b1, b2, b3;

B3 = b1+b2;

A class is made abstract by declaring at least one of its functions as pure virtual function. A pure virtual function is specified by placing "= 0"

The purpose of an abstract class (often referred to as an ABC) is to provide an appropriate base class from which other classes can inherit. Abstract classes cannot be used to instantiate objects and serves only as aninterface. Attempting to instantiate an object of an abstract class causes a compilation error.

Classes that can be used to instantiate objects are called concrete classes.