Beginning Yesterday In Your my.ucla.edu Portal, You Can Complete Student Evaluations Of This Course. I Appreciate The Feedback And Constructive Criticism. They Are Not Released To Me Until I Post Final Letter Grades.

Project 5 Is Coming Due Tonight At 9PM. You Have Until Thursday At 7 AM To Submit Late Work. I Am Flying To Seattle Late This Afternoon So I Won't Be All That Available Until After 8 PM. Just So You Are Aware…

Final Exam Information: Scheduled For Wednesday August 15 - 10-Noon. I Will Be Releasing Final Exam Study Problems That We Will Discuss In Class Next Week. Please Review My Solution To Projects 3 and 4 and 5 and 6. You Can Bring In Two 8 1/2 x 11 Sheets Of Whatever Notes You Like. Both Sides. Typed Or Scribbled…

8 Questions. Out of 100
About 30 M/C T/F Kinds Of Questions
Something Involving Project 6
Something Related To Project 3
Something Related To Project 5
Build A Class .h and .cpp
A Small String Problem
A Small Array Problem
Parts Of The Coded Solutions To Project 3, 4 And 5 To Change

- Project 6

Gosh really, Howard why are you so enamored with enumerations?
Enum GAMEOUTCOME { HUMANWON, COMPUTERWON=4, TIEDGAME, NOTYETDECIDED };  
/// fixed set known at compile-time that I can name  
/// once named, they never ever change....  

/// cram a number in there...  
GAMEOUTCOME o = Bunco::GAMEOUTCOME::3;  /// not recommended  
GAMEOUTCOME o = Bunco::TIEDGAME;  

Class Person  
{  
Public:  
    Person();  
    Person( string name, string address );  

Private:  
    string mName;  
    string mAddress;  
};  

Person::Person( string name, string address )  
    : mName( name ), mAddress( address )  
{  
    // body of your constructor empty...  
}
Class Student : public Person

// layers of an onion  additive model
{  /// special  make a Student different person
Public:
  Student( string ID, string name, string address );

Private:
  string mID;
};

Student::Student( string id, string name, string address )
  : Person( name, address ), mID( id )
{
  // empty…
  // Person( name, address );
}

Int I = 17;
Int * ptrI = & I;   /// aliases together…
ptrI = nullptr;
ptrI = new int( 18 );
delete( ptrI );
Int  j = 20;
ptrJ = j;
ptrI = ptrJ;

Int array[ 5 ];
array = ptrI;

ptrI ----> int *

**ptrI = array;  // legal?? Work?? Mean? ptrI array[0]**

array = ptrI;  /// ILLEGAL

array variable address of a blob
fixed unmoveable address
Don't try to move your array variable

Array ----> int * const

Array + 1