#### Q1: Function to find & display sum of rows & sum of cols. of a 2 dim. array A

####

#include <iostream>

using namespace std;

void SumRowCol(int A[][3], int , int );

int main()

{

 int A[4][3] ={{12, 29, 11},

 {25, 25, 13},

 {24, 64, 67},

 {11, 18, 14}};

 SumRowCol(A,4,3);

 return 0;

}

void SumRowCol(int A[][3], int N, int M)

{

 for(int R=0;R<N;R++)

 {

 int SumR=0;

 for(int C=0;C<M;C++)

 SumR+=A[R][C];

 cout<<"Row("<<R<<")="<<SumR<<endl;

 }

 for(int C=0;C<M;C++)

 {

 int SumC=0;

 for(int R=0;R<N;R++)

 SumC+=A[R][C];

 cout<<"COL("<<C<<")="<<SumC<<endl;

 }

}

Q2) // Selection sort with pass-by-reference. This program puts values into an

// array, sorts them into ascending order and prints the resulting array.



#include <iostream>

using namespace std;

#include <iomanip>

using std::setw;

void selectionSort( int \* const, const int ); // prototype

void swap( int \* const, int \* const ); // prototype

int main()

{

 const int arraySize = 10;

 int a[ arraySize ] = { 2, 6, 4, 8, 10, 12, 89, 68, 45, 37 };

 cout << "Data items in original order\n";

 for ( int i = 0; i < arraySize; i++ )

 cout << setw( 4 ) << a[ i ];

 selectionSort( a, arraySize ); // sort the array

 cout << "\nData items in ascending order\n";

 for ( int j = 0; j < arraySize; j++ )

 cout << setw( 4 ) << a[ j ];

 cout << endl;

 return 0;

}

// function to sort an array

void selectionSort( int \* const array, const int size )

{

 int smallest; // index of smallest element

 // loop over size - 1 elements

 for ( int i = 0; i < size - 1; i++ )

 {

 smallest = i; // first index of remaining array

 // loop to find index of smallest element

 for ( int index = i + 1; index < size; index++ )

 if ( array[ index ] < array[ smallest ] )

 smallest = index;

 swap( &array[ i ], &array[ smallest ] );

 } // end if

} // end function selectionSort

// swap values at memory locations to which

// element1Ptr and element2Ptr point

void swap( int \* const element1Ptr, int \* const element2Ptr )

{

 int hold = \*element1Ptr;

 \*element1Ptr = \*element2Ptr;

 \*element2Ptr = hold;

}

Q3) Data Members, set Functions and get Functions



// Define class GradeBook that contains a courseName data member

// and member functions to set and get its value;

// Create and manipulate a GradeBook object.

#include <iostream>

using std::cout;

using std::cin;

using std::endl;

#include <string> // program uses C++ standard string class

using std::string;

using std::getline;

// GradeBook class definition

class GradeBook

{

public:

 // function that sets the course name

 void setCourseName( string name )

 {

 courseName = name; // store the course name in the object

 } // end function setCourseName

 // function that gets the course name

 string getCourseName()

 {

 return courseName; // return the object's courseName

 } // end function getCourseName

 // function that displays a welcome message

 void displayMessage()

 {

 // this statement calls getCourseName to get the

 // name of the course this GradeBook represents

 cout << "Welcome to the grade book for\n" << getCourseName() << "!"

 << endl;

 } // end function displayMessage

private:

 string courseName; // course name for this GradeBook

}; // end class GradeBook

// function main begins program execution

int main()

{

 string nameOfCourse; // string of characters to store the course name

 GradeBook myGradeBook; // create a GradeBook object named myGradeBook

 // display initial value of courseName

 cout << "Initial course name is: " << myGradeBook.getCourseName()

 << endl;

 // prompt for, input and set course name

 cout << "\nPlease enter the course name:" << endl;

 getline( cin, nameOfCourse ); // read a course name with blanks

 myGradeBook.setCourseName( nameOfCourse ); // set the course name

 cout << endl; // outputs a blank line

 myGradeBook.displayMessage(); // display message with new course name

 return 0; // indicate successful termination

} // end main