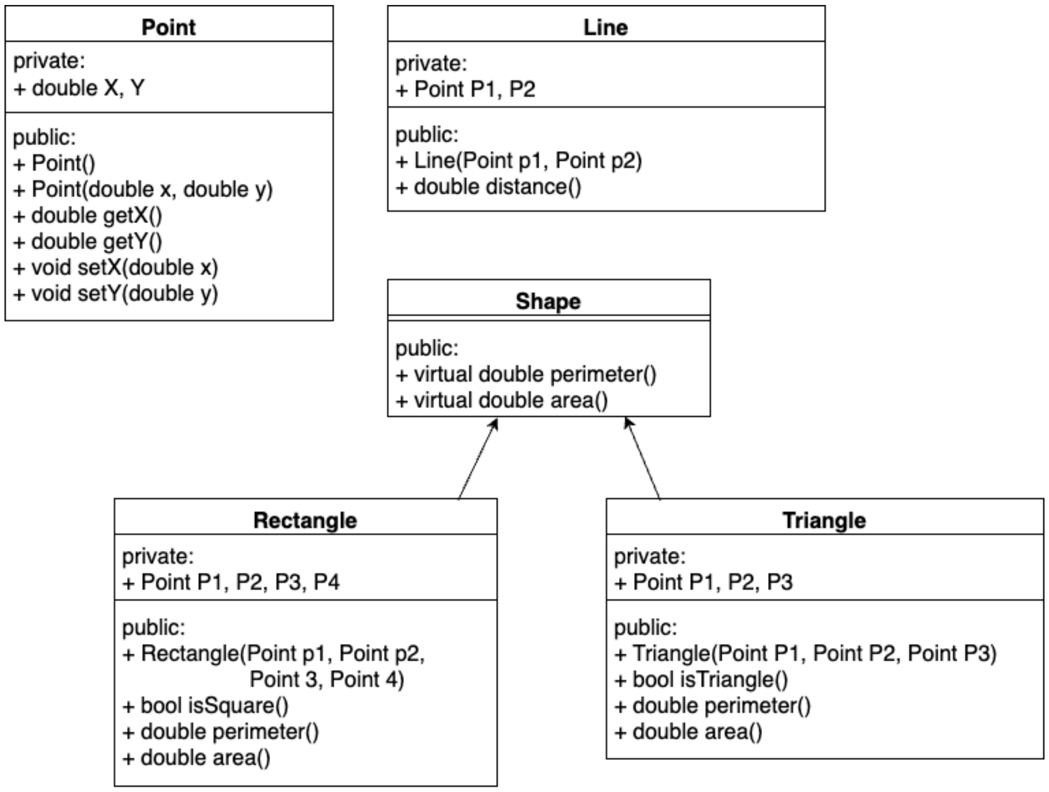
Midterm Examination

DURATION: 120 minutes

- Please write your program and save your files with name as follow:
  [student_id]_p[question_number].cpp Example: s123456_p1.cpp
- Put all files into a folder as your student ID to help TAs collect your files.
- Please double check and make sure your files have been collected before leaving.

Question 1. (30 points) Given the following UML:



Requirements:

1. (20 points) Define class **Point, Line, Shape, Rectangle** and **Triangle** follow the design. Class **Rectangle** and **Triangle** are derived classes of **Shape.** Here is the formular to calculate distance between two points.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

2. (10 points)
   a. Function **isTriangle()** return true if these 3 points form a triangle. Hint: The sum of the lengths of any two sides is greater than the remaining side. Otherwise, return false.

b. Function **isSquare()** return **true** if these 4 edges are equal. Otherwise, return **false**.

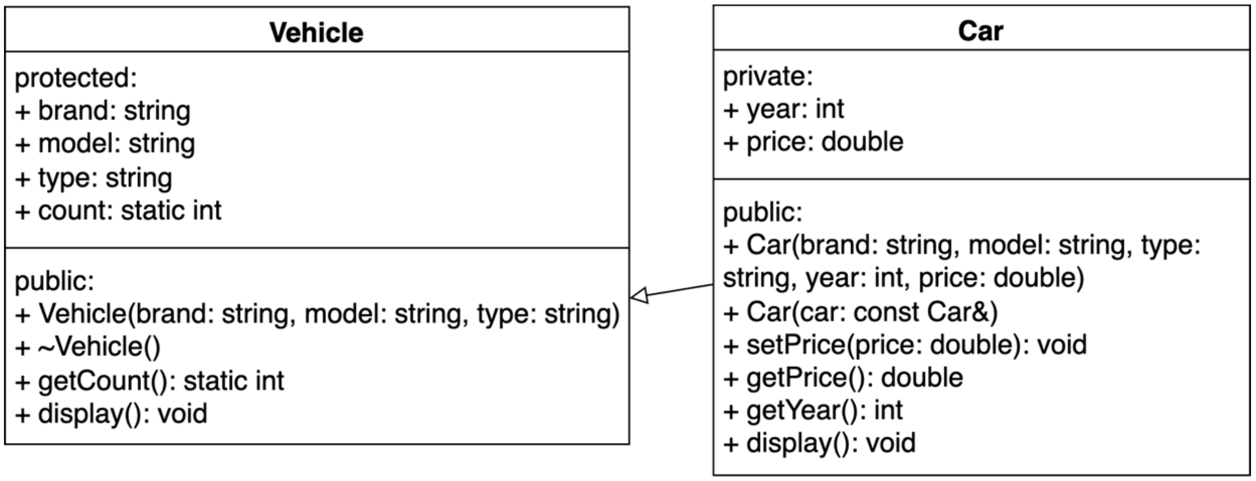
c. In case you don't remember how to calculate the area of triangle with 3 sides. I provide you Heron's formula here.

$$\text{Area, } A = \sqrt{s(s-a)(s-b)(s-c)}$$

Where,

$$S = \text{Semi Perimeter} = \frac{a+b+c}{2}$$

Question 2. (40 points) Given the following UML:



Requirements:

3. (10 points) Define class **Vehicle** follow the design:
    a. The brand, model, year data members are **protected**, Vehicle(…) is constructor, ~Vehicle() is destructor.
    b. The static variable count will increase whenever a new vehicle created, decrease whenever a vehicle deleted. The static method getCount returns number of vehicles created.
    c. The display method in Vehicle class will output detail of the vehicle in one line.
4. (10 points) Define class **Car** follow the design:
    a. The Car class inherits Vehicle class, The year, price data members are **private**. The Car class also have one constructor and one copy constructor.
    b. The setPrice method will assign a non-negative number to the data member. If the parameter is negative number, assign to 0.
    c. The display method in Car class will output detail of the car in one line.
5. (20 points) Create a main function as follow:

a. (5 points) Create an object of Vehicle class with any detail by using constructor and call display method to output the detail of vehicle to the screen.
b. (5 points) Create array of objects from Car class follow this table, then print out the detail of cars and total number of vehicle by using getCount method.
c. (10 points) Delete vehicle object in requirement 3a, then output the detail of cars **in ascending price order** and total number of vehicles by using `getCount` method.

Model	Brand	Type	Year	Price
Toyota	Camry	sedan	2015	9800
Ford	Escape	crossover	2015	15900
Honda	Civic	sedan	2016	10200
Toyota	RAV4	crossover	2016	12800
Toyota	4Runner	suv	2015	16900
Honda	CR-V	crossover	2016	17900

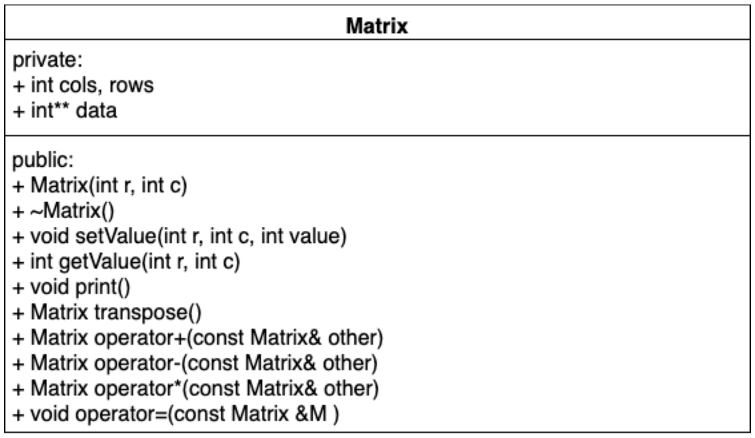
Please follow the input/output as below:

```
Brand: Volkswagen    Model: Golf    Type: compact

> LIST OF CARS:
Model    Brand         Type          Year    Price
==================================================
Toyota   Camry         sedan         2015    9800
Ford     Escape        crossover     2015    15900
Honda    Civic         sedan         2016    10200
Toyota   RAV4          crossover     2016    12800
Toyota   4Runner       suv           2015    16900
Honda    CR-V          crossover     2016    17900
==================================================
Total vehicle: 7

> DELETE FIRST VEHICLE SUCCESSFULLY!
> LIST OF CARS IN ASCENDING PRICE ORDER:
Model    Brand         Type          Year    Price
==================================================
Toyota   Camry         sedan         2015    9800
Honda    Civic         sedan         2016    10200
Toyota   RAV4          crossover     2016    12800
Ford     Escape        crossover     2015    15900
Toyota   4Runner       suv           2015    16900
Honda    CR-V          crossover     2016    17900
==================================================
Total vehicle: 6
```

Question 3 (30 points) Create class Matrix as below UML design:



```
Matrix
private:
+ int cols, rows
+ int** data
public:
+ Matrix(int r, int c)
+ ~Matrix()
+ void setValue(int r, int c, int value)
+ int getValue(int r, int c)
+ void print()
+ Matrix transpose()
+ Matrix operator+(const Matrix& other)
+ Matrix operator-(const Matrix& other)
+ Matrix operator*(const Matrix& other)
+ void operator=(const Matrix &M )
```

Explanation:

Constructor **Matrix(int r, int c)** will allocate data for the matrix by using pointer.

data = new int*[rows]   for each row: data[i] = new int[cols]

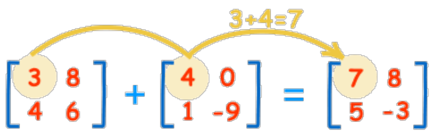
Destructor **~Matrix()** will delete memory space for data use delete.

Hint: Transpose matrix

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{2 \times 3} \qquad A^T = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}_{3 \times 2}$$

The sum of two matrices A and B is matrix C calculated by the formula:

$$c_{ij} = a_{ij} + b_{ij}\ (i = 0,1,2,\dots,m-1, j = 0,1,2,\dots,n-1)$$

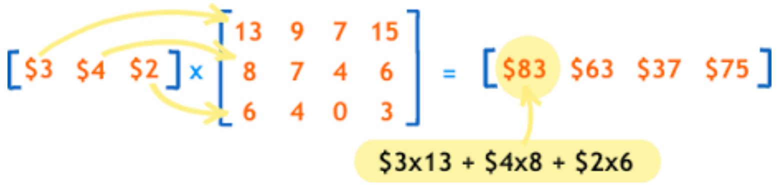


$$\begin{bmatrix} 3 & 8 \\ 4 & 6 \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 1 & -9 \end{bmatrix} = \begin{bmatrix} 7 & 8 \\ 5 & -3 \end{bmatrix}$$

The product of two matrices A and B is matrix C calculated by the formula:

$$c_{ij} = a_{i1} * b_{1j} + a_{i2} * b_{2j} + a_{i3} * b_{3j} + \cdots + a_{ik} * b_{kj}$$

$$(i = 0,1,2,\dots,m-1, j = 0,1,2,\dots,n-1)$$


[$3 $4 $2] x [13 9 7 15 ; 8 7 4 6 ; 6 4 0 3] = [$83 $63 $37 $75]
$3x13 + $4x8 + $2x6

Here is the main function and sample output for testing:

```
int main() {
    Matrix mat1(2, 3);
    Matrix mat2(2, 3);

    // Initialize matrices
    mat1.setValue(0, 0, 1);
    mat1.setValue(0, 1, 2);
    mat1.setValue(0, 2, 3);
    mat1.setValue(1, 0, 4);
    mat1.setValue(1, 1, 5);
    mat1.setValue(1, 2, 6);
    mat2.setValue(0, 0, 7);
    mat2.setValue(0, 1, 8);
    mat2.setValue(0, 2, 9);
    mat2.setValue(1, 0, 10);
    mat2.setValue(1, 1, 11);
    mat2.setValue(1, 2, 12);

    // Perform operations
    Matrix sum = mat1 + mat2;
    Matrix diff = mat2 - mat1;
    Matrix prod = mat1 * mat2.transpose(); // This will throw an error since dimensions are not compatible
    Matrix trnsp = mat1.transpose(); // This will throw an error since transpose is not defined

    // Print the results
    cout << "Matrix A:" << endl;
    mat1.print();
    cout << "Matrix B:" << endl;
    mat2.print();
    cout << "Transpose of Matrix A:" << endl;
    trnsp.print();
    cout << "Matrix A + Matrix B:" << endl;
    sum.print();
    cout << "Matrix B - Matrix A:" << endl;
    diff.print();
    cout << "Matrix A * transpose(Matrix B):" << endl;
    prod.print();

    return 0;
}
```

SAMPLE OUTPUT

```
Matrix A:
1 2 3
4 5 6
Matrix B:
7 8 9
10 11 12
Transpose of Matrix A:
1 4
2 5
3 6
Matrix A + Matrix B:
8 10 12
14 16 18
Matrix B - Matrix A:
6 6 6
6 6 6
Matrix A * transpose(Matrix B):
50 68
122 167
```