Section, Name and ID#:____

AMERICAN UNIVERSITY OF ARMENIA
College of Science and Engineering
CS 120 Introduction to Object-Oriented Programming

QUIZ 09

5

Date / Time:	Friday, April 14 2017 at 17:00
Duration:	1 hour
Attention:	**ANY TYPE OF COMMUNICATION IS STRICTLY PROHIBITED**
	Write down your section, name and ID# at the top of all used pages
Instructions:	1. Write the solutions in the top half of each page under problem statements
	2. Copy the same solution in the bottom section to take with you after quiz
	3. Turn your solution into a program, compile and submit the errors
	4. Correct the errors and submit the working version of your program
Submission Deadline:	**Sunday, April 16 2017, before 22:00**
Submission Contact:	**skhachat@aua.am**
	arshavir.voskanyan@gmail.com, nareh_salmasian@edu.aua.am

Problem 1: A rectangle with sides parallel to *x* and *y* axes can be represented by its diagonal of type ***line***. Implement a C++ class rectangle (its member functions) assuming the existence of all necessary functions of the class line:

```
class rectangle
{ public:
      rectangle(double x0, double y0, double x1, double y1); // initializes by
                                                      //bottom-left and top-right coordinates
      double perimneter();
      double area();
      bool intersect(rectangle &that); // checks if the rectangles intersect
      rectangle union(rectangle &that); // returns least rectangle that includes both
  private:
      line diagonal; // ~~arrays of x and y coordinates of vertices respectively~~
}
```

1

```
                                                                         :diagonal()
rectangle::rectangle ( double  x0, double y0 , double  x1 ,double y1)
{  x0= diagonal. get_x0();
   y0= diagonal.get_y0();
   x1 = diagonal. get_x1();
   y1 = diagonal . get_y1(); }
double rectangle:: perimeter(double a; double b) {
  a= y1- y0;
  b= x1-x0;
  return 2* (a+b); }
 double rectangle::area () {
          return  a*b; }.
```

```
bool rectangle:: intersect
 (rectangle  & that) {
line intersect (rectangle  & that)}.
```

Student's copy

OOP.Q09.140417.H051

Use the backside, if needed

Problem 1 of 3

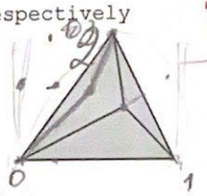
Section, Name and ID#:____

Problem 2: Implement a C++ ***class triangle*** (only its member functions marked by **TODO**) the header file of which is given below. The Heron's formula is $area = \sqrt{p(p-a)(p-b)(p-c)}$, where p is the half-perimeter and ***a***, ***b*** and ***c*** are the sides.

```
class triangle
{ public:
      triangle(double vertex[][3]); // TODO - initializes vertices by specified
                                    // array of two rows and three columns
      double get_x(int vertex); // returns x coordinate of specified vertex
      double get_y(int vertex); // returns y coordinate of specified vertex
      double side(int vertex); // returns side length from specified vertex to next one

      double perimneter(); // TODO
      double area(); // TODO - computes area using Heron's formula
      bool is_inside(double px, double py); // TODO - checks if a point  with coordinates
                            // (px, py) is inside the triangle - see shaded areas below
   private:
      double x[3], y[3]; // arrays of x and y coordinates of vertices respectively
}
```

3



```
triangle::triangle( double vertex [2][3]) {
  for (int i=0, i<3; i++)
     x[i] = vertex [0][i];
     y[i]= vertex [1][i]; }

double triangle::perimeter() { return
double perimeter = side(0) +
 + side(1) + side(2) ; }.

double triangle:: get_x (int vertex) {
    return   x [vertex]; }

double triangle area() {
return double area= sqrt ( perimeter()/2*
 * ( perimeter()/2 - side (0))*(permeter (0)/2-side(1))*
   *(permeter()/2- side(2)) ; }.

double triangle :: get_y (int vertex){
   return  y [vertex]; }.

double triangle::side (int vertex) {double side;
   if  ( vertex <0|| vertex > 2)
    return 0 ;
   else if (vertex = 2)
return  ~~i~~side =  sqrt ( (x [2]- x[0])*(x[2]-x[0])+ (y[2]-y[0])*( y[2]-y[0]);
   else
return. side = sqrt ( (x [vertex+1] - x[vertex])*(x[vertex+1]-x(vertex]) + (y[vertex+1] -
   -y(vertex]) )* ( y [vertex+1]- y [vertex]) ; }

bool triangle :: is_inside(double px, double py)
{  if (px >=get_x(0)|| px <= get_x(1) &&
   (py >= get_y(0) || py <=get_y(2))
      return 1; else  return 0; }
```

Student's copy

OOP.009.140417.H051

Use the backside, if needed

Section, Name and ID#:____

Problem 3: Write and implement the following C++ classes:

1. ***class course*** – encapsulates three member variables ***string name, int crtedits*** and ***double grade.***
2. ***class semester*** – encapsulates a six-element private array course subjects[6] and implements ***void set(int i, string new_name, int new_units, double new_grade), course get(int i), int total_credits()*** and ***double gpa()*** public functions. If total credits are ***0***, the gpa is also ***0***. Make appropriate changes in ***class course.***

```
class course {                       Const-?                 1
public: string name;
     int credits;
   double grade; }.
class semester {
private:
  course  subjects [6];
 public:
void set (int i, string new_name, int new_units, double new_grade);
course get (int i);
int total_credits();
double gpa(); }
  semester:: semester ()
{                                  }.
semester:: void set (int i, string new_name, int new_units, double new_grade) {
return   name = new_name;              | int semester total_credits() {
return   credits = new_units;          | for (int i=0; i< 6; i++)
return      grade = new_grade; }       |  return
course  get (int i) {                  |
       name = course subjects[i]; }    |
return needed
```

Student's copy

OOP.Q09.140417.H051

Use the backside, if needed