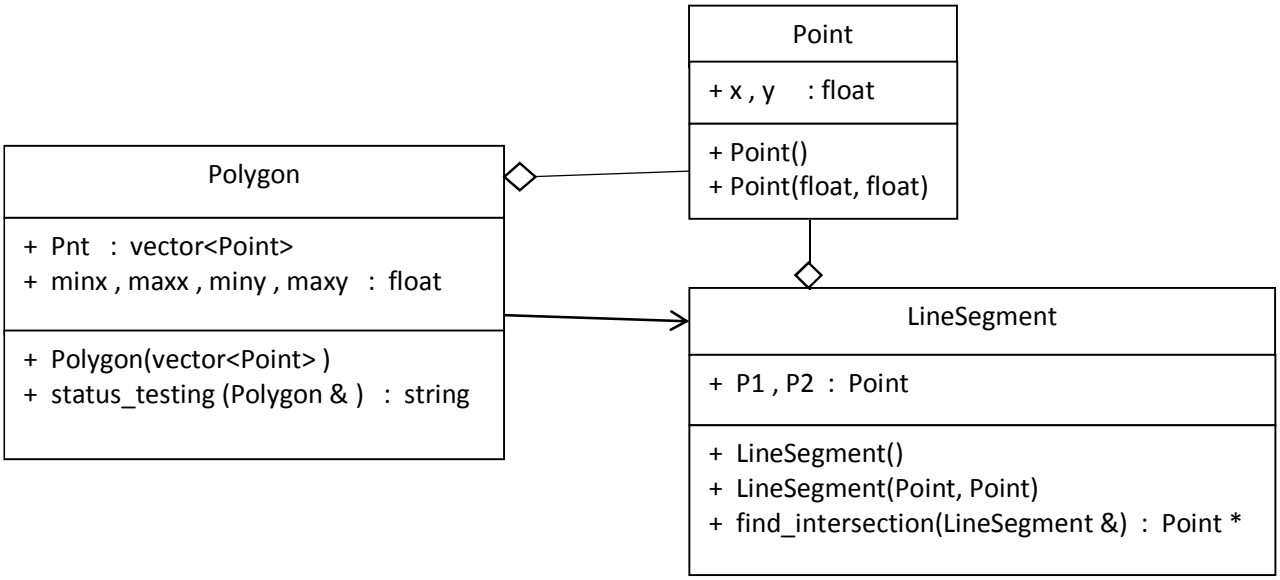
BLG252E – Object Oriented Programming
SAMPLE FINAL EXAM
Duration: 2 hours. (Books and notes closed.)

Use the following UML class diagram for all questions, except the last question.



In your answers, you should use the **STL (Standard Template Library) vector** objects whenever indicated in UML diagram.

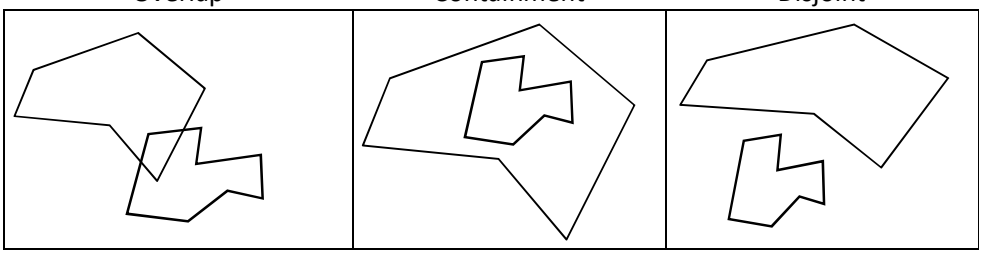
QUESTION 1) [5 points] Write C++ codes for the **Point** class.

x and y	Two float values represent the point in coordinate system.
Default constructor	Does nothing.
Parametered constructor	Takes float values for x and y members.

QUESTION 2) [30 points] Write C++ codes for the **LineSegment** class.

P1 and P2	P1 represents the start point, P2 represents the end point of the line segment.
Default constructor	Does nothing.
Parametered constructor	Takes Point objects for P1 and P2 members.
find_intersection() function	Takes another LineSegment object as input parameter, then checks whether there is an intersection between itself and the given LineSegment. If there is no intersection, function returns NULL. Otherwise function constructs a newly allocated Point object (by using x and y values of intersection), then returns a pointer to that new Point object. ▪ A line which passes from two known points is represented by the following equation: y=m.x+b , where m is the slope ($m = \frac{y_2-y_1}{x_2-x_1}$) and b is the shift ($b = \frac{x_2y_1-x_1y_2}{x_2-x_1}$) ▪ If a slope can not be calculated due to division-by-zero, then function should display a warning message and program must stop. ▪ If m_1 and m_2 are equal, then you can conclude that two lines are parallel and they do not intersect. Otherwise, they potentially intersect. The x intersection coordinate of two lines is calculated by $x_{intersection} = \frac{b_2-b_1}{m_1-m_2}$. ▪ Finaly check whether the calculated intersection point ($x_{intersection}$ and $y_{intersection}$) is within the borders of both line segments. If this condition is true, then function should return a pointer to the intersection Point object, otherwise function should return NULL.

QUESTION 3) [40 points] Write C++ codes for the **Polygon** class.

Pnt vector	Pnt is a **STL vector** of Point objects. Each Point object in vector represents a vertex (corner) in the polygon. Two consequtive Points are considered to be connected in clock-wise sequence. The last Point in vector (***(N-1)th)*** is considered to be connected to the first Point ***(0th)***, so that the polygon is closed properly.
minx, maxx, miny, maxy	These values represent the extremites in the polygon : minx (left most) , maxx (right most), miny (bottom most), maxy (top most).
Parametered constructor	Takes a vector of Point objects for Pnt member. Also, function determines and assigns the extremite values (minx, maxx, miny, maxy) by using the Pnt vector. The extremite values can be used for testing the "Containment" status.
status_ testing() function	Takes another Polygon object, then tests the status between itself and the given Polygon. Function should return one the following strings: "Overlap", "Containment", or "Disjoint". The followings are status examples of two polygons. Overlap Containment Disjoint  ▪ Overlap testing: Build **LineSegment** objects of the two polygons and check whether any line segments intersect. If there is at least one intersection, then you can conclude status is "overlap". ▪ Containment testing: Check if there is any "containment" status by using the extremite values of the two polygons. ▪ Disjoint testing: If neither an Overlap nor a Containment found, then you can conclude status is "disjoint".

QUESTION 4) [10 points] Write the main program to do followings:

1. Declare an array (A1) of Point objects,initializing with following points : (1, 3) , (2, 5) (4, 6) (6, 4) (5, 1) (3, 4)
2. Declare an array (A2) of Point objects,initializing with following points : (2, 1) , (4, 3) (7, 2.5) (6, -2)
3. Declare a Polygon object (Pol1), invoking its constructor with a Point vector that you construct from A1.
4. Declare a Polygon object (Pol2), invoking its constructor with a Point vector that you construct from A2.
5. Test the status of Pol1 with Pol2 and display the result on screen.

QUESTION 5) [15 points] What is the screen output of the following program?

```
class A {
int i, j, k;
public:
A(int x=1) : i(x), j(2), k(3)  {cout << "\nA constructor ";print(); }
void print() {cout <<i<<" "<<j<<" "<<k<<" , "; }
~A() {cout << "A destructor\n";}   };

class B : public A {
int i, j, k;
public:
B(int x=1, int y=2) : A(x), i(y), j(3), k(4)
  {cout << "\nB constructor ";print(); }
void print() {A::print(); cout <<i<<" "<<j<<" "<<k<<" , "; }
~B() {cout << "B destructor\n";}   };
```

```
class C : public B {
int i, j, k;
public:
C(int x=1, int y=2, int z=3) : B(x, y), i(z), j(4), k(5)
  {cout << "\nC constructor ";print(); }
void print() {B::print(); cout <<i<<" "<<j<<" "<<k<<" , "; }
~C() {cout << "C destructor\n";}   };

int main() {
  cout << "main start\n";
  C  obj(10, 20, 30);
  cout << "main end\n";
  return 0;
}
```