

# Homework # 3 (Due Date: Thursday December 12, 2013)

## Submission:

Submit your homework via [E-learning](#)

## Deliverables:

5 code files (main.cpp, Point.cpp, Point.h, Polygon.cpp, and Polygon.h)

4 images (test1.jpg, test2.jpg, test3.jpg, and test4.jpg)

Select all 9 files and compress them into a file named **StudentID\_LastName\_Section#.zip**

Note: use [Winrar](#) to compress your files.

## Description:

You are going to implement two classes **Point** and **Polygon**.

Point class used to represent points in 3D space. Each point has x, y, and z coordinates to represent point position in space.

Polygon class used to represent shapes specified by 3 or more points.

```
class Point
{
    // Point member functions
private:
    double x, y, z;
};

class Polygon
{
    // Polygon member functions
private:
    int numOfPoints;
    Point* points;
};
```

**Tasks:** Implement each of the following functions

### Class Point

#### 1. Constructor

```
// parameters: 3 double parameters a, b, and c
// return: none
```

takes 3 double parameters a, b, and c to set the values of x, y, and z. If no parameters are passed, a, b, and c take a default value of 0.0

#### 2. GetPoint

```
// parameters: 3 double parameters a, b, and c by reference
// return: none
```

takes 3 double parameters a, b, and c by reference to return the values of x, y, and z.

#### 3. SetPoint

```
// parameters: 3 double parameters a, b, and c
// return: none
```

takes 3 double parameters a, b, and c to set the values of x, y, and z.

#### 4. Print

```
// parameters: none
// return: none
```

prints the values of x, y, and z in the form (x, y, z).

### Class Polygon

#### 1. Default Constructor

```
// parameters: none
// return: none
```

creates an empty polygon that has no points by setting numOfPoints to 0 and points to NULL.

## 2. Constructor

```
// parameters: an integer num and a pointer to array of points ps  
// return: none
```

creates a polygon that has **num** points

if **num** < 3, the operation can not be proceeded, so set numOfPoints to 0, points to NULL and exit the function otherwise

set numOfPoints to **num**

allocate memory for points and copy the contents of **ps** to points.

## 3. Copy Constructor

```
// parameters: a constant Polygon object by reference  
// return: none
```

creates a polygon that is an exact copy of another polygon passed to the constructor

## 4. Destructor

```
// parameters: none  
// return: none
```

deallocates the dynamic memory of points and sets it to NULL.

## 5. SetPolygon (almost same code as the constructor above)

```
// parameters: an integer num and a pointer to array of points ps  
// return: none
```

creates a polygon that has **num** points

if points != NULL (points already has some points), delete points memory and set numOfPoints to 0

if **num** < 3, the operation can not be proceeded, so set numOfPoints to 0, points to NULL and exit the function otherwise

set numOfPoints to **num**

allocate memory for points and copy the contents of **ps** to points.

## 6. Destroy (same code as the destructor above)

```
// parameters: none  
// return: none
```

deallocates the dynamic memory of points and sets it to NULL.

## 7. Print

```
// parameters: none  
// return: none
```

prints how many points in the polygon and all its points.

### Testing:

Upon the completion of your implementation, you need to conduct 4 test cases (see below) to test the correctness of the functionality of your implementation. Design a main function for each test case, run the code and save an image of the debug window. 4 image files (test1.jpg, test2.jpg, test3.jpg, and test4.jpg) must be submitted along with your code files.

### How to save an image of the debug window

while the debug window is showing, click alt + print screen keys on your keyboard

open MS paint, and click ctrl + v keys to paste the image

save the image in JPEG format

### Test case 1: create an empty polygon

```
void main ()  
{  
    cout<<"+-----\n";  
    cout<<"My Information:\n\tStudent Name\n";  
    cout<<"\tStudent ID\n";  
    cout<<"\tSection #\n";  
    cout<<"+-----\n\n";  
  
    Polygon p1;
```

```

    p1.Print ();
}

```

```

C:\Windows\system32\cmd.exe
My Information:
    Student Name
    Student ID
    Section #
Polygon has NO Points
Press any key to continue . . . .

```

**Test case 2:** an attempt to create a polygon with only 2 points will not succeed

```

void main ()
{
    cout<<"-----\n";
    cout<<"My Information:\n\tStudent Name\n";
    cout<<"\tStudent ID\n";
    cout<<"\tSection #\n";
    cout<<"-----\n\n";

    int num = 2;
    Point* points = new Point[num];
    points[0].SetPoint (10, 5, -7);
    points[1].SetPoint (-3, -4, 0);
    Polygon p2 (num, points);
    p2.Print ();

    delete [] points;
}

```

```

C:\Windows\system32\cmd.exe
My Information:
    Student Name
    Student ID
    Section #
Polygon has NO Points
Press any key to continue . . . .

```

**Test case 3:** create a polygon with 5 points  
make a copy of the polygon then destroy the old polygon

```

void main ()
{
    cout<<"-----\n";
    cout<<"My Information:\n\tStudent Name\n";
    cout<<"\tStudent ID\n";
    cout<<"\tSection #\n";
    cout<<"-----\n\n";

    int num = 5;
    Point* points = new Point[num];
    double x, y, z;
    for (int i = 0; i < num; ++i) {
        cout<<"Enter x, y, and z coordinates:";
        cin>>x>>y>>z;
        points[i].SetPoint (x, y, z);
    }
    Polygon p3 (num, points);

    Polygon p4 (p3);

    p3.Destroy ();

    p3.Print ();

    p4.Print ();

    delete [] points;
}

```

```

C:\Windows\system32\cmd.exe
My Information:
    Student Name
    Student ID
    Section #
Enter x, y, and z coordinates:1 2 3
Enter x, y, and z coordinates:-0.5 6.7 10
Enter x, y, and z coordinates:30 20 -0.3
Enter x, y, and z coordinates:20 0 0
Enter x, y, and z coordinates:4 5 9
Polygon has NO Points
Polygon has 5 Points
Point 0 : <1, 2, 3>
Point 1 : <-0.5, 6.7, 10>
Point 2 : <30, 20, -0.3>
Point 3 : <20, 0, 0>
Point 4 : <4, 5, 9>
Press any key to continue . . . .

```

**Test case 4: (write the code yourself)**

create a polygon p5 with 3 points  
print p5  
use SetPolygon member function to make p5 have 6 points  
print p5

```
C:\Windows\system32\cmd.exe
My Information:
Student Name
Student ID
Section #

Enter x, y, and z coordinates:10 20 30
Enter x, y, and z coordinates:20 0 0
Enter x, y, and z coordinates:30 -20 -50

Polygon has 3 Points
Point 0 : <10, 20, 30>
Point 1 : <20, 0, 0>
Point 2 : <30, -20, -50>
Enter x, y, and z coordinates:1 2 3
Enter x, y, and z coordinates:5 6 7
Enter x, y, and z coordinates:8 9 10
Enter x, y, and z coordinates:0 0 0
Enter x, y, and z coordinates:-9 10 5
Enter x, y, and z coordinates:5.5 6 0

Polygon has 6 Points
Point 0 : <1, 2, 3>
Point 1 : <5, 6, 7>
Point 2 : <8, 9, 10>
Point 3 : <0, 0, 0>
Point 4 : <-9, 10, 5>
Point 5 : <5.5, 6, 0>
Press any key to continue . . .
```