

UNIVERSITY OF BURGUNDY

SOFTWARE ENGINEERING

TUTORIAL 4

Lab Report-4

Author:

KODIPAKA VAMSHI

Supervisor:

Dr. Yohan FOUGEROLLE

November 5, 2018



1 Point2D

```
#include <iostream>
using namespace std;
struct Point2D
{
    //    char name[50];
    float x;
    float y;
};
void display(Point2D p1){
    //    cout << "Point Name: " << p1.name << endl;
    cout << "x:_" << p1.x << endl;
    cout << "y:_" << p1.y << endl;
}
void display(Point2D);
void display1(Point2D (*ptr)){
    //    cout << "Point Name: " << (*ptr).name << endl;
    cout << "x:_" << (*ptr).x << endl;
    cout << "y:_" << (*ptr).y << endl;
}
void display1(Point2D);

    void BuildPoint(Point2D);
void BuildPoint(Point2D (*p2)){
    //    cout << "Enter Point name: ";
    //    cin.get(p2->name, 50); cout << endl;
    cout << "Enter_x_coordinate:_" ;
    cin >> (p2->x);
    cout << "Enter_y_coordinate:_" ;
    cin >> (p2->y); cout << endl;
}
```

```
main()
{
    Point2D *ptr, p1, *ptr2, p2;
    // Point2D dummy={ "a", 10, 9 };
    Point2D dummy;
    // dummy.name[50] = 'dummy';
    dummy.x = 10.3;
    dummy.y = 9.88;
    cout << "Enter_Point_Information:_" << endl;
    //cin.get(p1.name, 50);
    cout << "Enter_x_coordinate:_" ;    cin >> p1.x;
    cout << "Enter_y_coordinate:_" ;    cin >> p1.y;
```

```

//display point2d-----
cout << "\nDisplaying Point Information." << endl;
display(p1);
cout << endl;
//display point2d by pointer1-----
ptr = &p1;
cout << "Displaying point info by pointers." << endl;
display1(ptr);
//cout << (*ptr).name << " " << (*ptr).x << " " << (*ptr).y << " ";
cout << endl;

ptr2 = &p2;
//Reading Point2D Information by pointer2-----
cout << "\nReading Point Info by pointer2."
_((BuildPoint::User_i/p_function)" << endl;
BuildPoint(ptr2);
cout << endl;
//display point2d by pointer2-----
cout << "Displaying point info by pointer2."
_((BuildPoint::User_i/p_function)" << endl;
display1(ptr2);
//cout << (*ptr).name << " " << (*ptr).x << " " << (*ptr).y << " ";
cout << endl;

cout << "Dummy structure is created already"<<endl;
// cout << " dummy Point name: " <<dummy.name<< endl;;
cout << " _dummy_x_coordinate:_"<<dummy.x<<endl;
cout << " _dummy_y_coordinate:_"<<dummy.y<<endl;
}

```

2 Polygon2d

```

#include <iostream>

using namespace std;

struct Polygon2D
{
    float x;
    float y;
    Polygon2D *prev;
    Polygon2D *next;
};

Polygon2D* BuildPolygon( int n)

```

```

{
    Polygon2D *Head = NULL;
    Polygon2D *Newnode=NULL,*temp=NULL;
    for(int i=0; i<n;i++)
    {
        Newnode = new Polygon2D;
        cout << "Enter x coordinate: "; cin>> Newnode->x;
        cout << "Enter y coordinate: "; cin>> Newnode->y;

        if (Head==NULL)
        {

            //intializing the node
            Head=Newnode;
            Newnode->next=Head;
            Newnode->prev=Head;

        }
        else
        {

            //intializing the node for second iteration

            Newnode-> prev = Head -> prev;
            Head -> prev -> next = Newnode;
            Newnode->next=Head;
            Head -> prev = Newnode;

        }
    }
    return Head;
}

void display(Polygon2D *Head, int n)
{
    Polygon2D *p=Head;
    for(int i=0; i<n;i++)
    {
        cout << "Current Points: " << endl;
        cout <<"x: " << p->x ;
        cout <<" ,y: " << p->y << endl;
        cout << "next Points: " << endl;
        cout <<"x: " << p->next->x ;
        cout <<" ,y: " << p->next->y << endl;
        cout << "Previous Points: " << endl;
    }
}

```

```

        cout <<"x:␣" << p->prev->x ;
        cout <<" ,y:␣" << p->prev->y << endl;
        p=p->next;
    }
}

```

```

int main()
{
    int n;
    cout<<"Enter the no of points:␣";
    cin>>n;
    Polygon2D *Head;
    Head = BuildPolygon(n);
    display(Head,3);
}

```

3 Polygon2D :: insert,delete a Point2D

```

#include <iostream>

using namespace std;

struct Point2D
{
    float x;
    float y;
    Point2D *prev;
    Point2D *next;
};

getelement(){
    Point2D *ptr, p1;
    cout << "Creation of a new Point:␣"<<endl;
    cout << "Enter x insert coordinate:␣";   cin >> p1.x;
    cout << "Enter y insert coordinate:␣";   cin >> p1.y;
    //display point2d by pointer1—————

    return(p1.x,p1.y);
}

Point2D* BuildPolygon( int n)
{

```

```

Point2D *Head = NULL;
Point2D *Newnode=NULL,*temp=NULL;
for(int i=0; i<n;i++)
{
    Newnode = new Point2D;
    cout << "Enter_x_coordinate:\n"; cin>> Newnode->x;
    cout << "Enter_y_coordinate:\n"; cin>> Newnode->y;

    if (Head==NULL)
    {
        //intializing the node
        Head=Newnode;
        Newnode->next=Head;
        Newnode->prev=Head;
    }
    else
    {
        //intializing the node for second iteration

        Newnode-> prev = Head -> prev;
        Head -> prev -> next = Newnode;
        Newnode->next=Head;
        Head -> prev = Newnode;
    }
}
return Head;
}

//void Readnode()
void display(Point2D *Head, int n)
{
    Point2D *p=Head;
    for(int i=0; i<n;i++)
    {
        cout << "Point_at_position:"<<i;
        cout<<endl;
        cout << "Current_Points:\n" << endl;
        cout <<"x:\n" << p->x ;
        cout <<" ,y:\n" << p->y << endl;
        cout << "next_Points:\n" << endl;
        cout <<"x:\n" << p->next->x ;
    }
}

```

```

        cout << ",y:_" << p->next->y << endl;
        cout << "Previous_Points:_" << endl;
        cout << "x:_" << p->prev->x ;
        cout << ",y:_" << p->prev->y << endl;
        p=p->next;
    }
}

void display1(Point2D (*ptr)){
    //    cout << "Point Name: " << (*ptr).name << endl;
    cout << "x:_" << (*ptr).x << endl;
    cout << "y:_" << (*ptr).y << endl;
}

void display1(Point2D);

Point2D BuildPoint(Point2D (*p2)){
    //    cout << "Enter Point name: ";
    //    cin.get(p2->name, 50); cout << endl;
    cout << "Enter_x_coordinate:_" ;
    cin >> (p2->x);
    cout << "Enter_y_coordinate:_" ;
    cin >> (p2->y); cout << endl;
    return (*p2);
}

getPosition(){
    int p;
    cout<<"enter_insert_location_of_point_in_polgon";
    cin>>p;
    return(p);
}

/* Point2D* InsertAt( Point2D *r, int c)
{
    r->prev=NULL;
    r->next=NULL;

    for(int i=0; i<c; i++)
    {

        if (Head==NULL)
        {

```

```

        //initializing the node
        Head=r;
        Newnode->next=Head;
        Newnode->prev=Head;

    }
    else
    {

        //initializing the node for second iteration

        r-> next = Head;
        Head -> prev= r;
        Head = r;

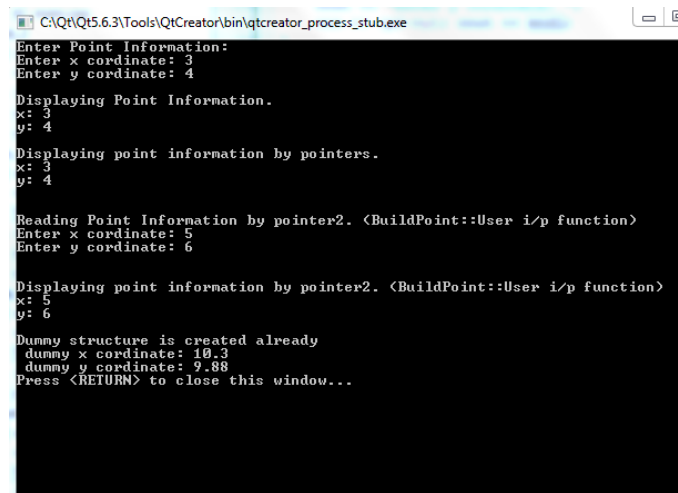
    }
}
return Head;
}

Point2D* DeleteAt( Point2D *r, int c){
    delete (*r);
};
*/

int main()
{
    int n,c;
    cout<<"Enter the no of points: ";
    cin>>n;
    Point2D *Head;
    Head = BuildPolygon(n);
    display(Head,3);

    Point2D *ptr2, p2,*r;
    ptr2 = &p2;
    //Reading Point2D Information by pointer2-----
    cout << "\nReading of Point (BuildPoint()) by pointer2." << endl;
    *r=BuildPoint(ptr2);
    cout << endl;
    //display point2d by pointer2-----
    cout << "Displaying point info by pointer2." << endl;
    display1(ptr2);
    //cout << (*ptr).name << " " << (*ptr).x << " " << (*ptr).y << " ";
    cout << endl;
    //enter point location to be inserted

```

```

C:\Qt\Qt5.6.3\Tools\QtCreator\bin\qtcreator_process_stub.exe
Enter Point Information:
Enter x coordinate: 3
Enter y coordinate: 4

Displaying Point Information.
x: 3
y: 4

Displaying point information by pointers.
x: 3
y: 4

Reading Point Information by pointer2. <BuildPoint::User i/p function>
Enter x coordinate: 5
Enter y coordinate: 6

Displaying point information by pointer2. <BuildPoint::User i/p function>
x: 5
y: 6

Dummy structure is created already
dummy x coordinate: 10.3
dummy y coordinate: 9.88
Press <RETURN> to close this window...

```

Figure 1: All outputs

```

c=  getPosition();
    /*  InsertAt(*r,c);
        //enter point location to be inserted
        DeleteAt(c);
    */
}

*****

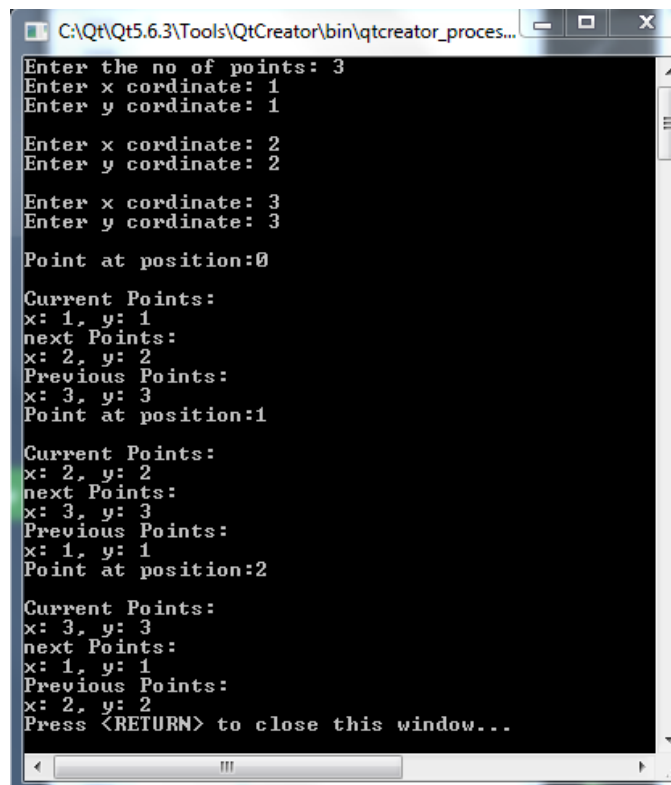
```

4 Outputs of all the Point2D

5 Outputs of all the Polygon2D

6 Outputs of insert and delete of Point2D in Polygon2D

6.1 Outputs of insert and delete are not displaying here



```
C:\Qt\Qt5.6.3\Tools\QtCreator\bin\qtcreator_proces...
Enter the no of points: 3
Enter x coordinate: 1
Enter y coordinate: 1

Enter x coordinate: 2
Enter y coordinate: 2

Enter x coordinate: 3
Enter y coordinate: 3

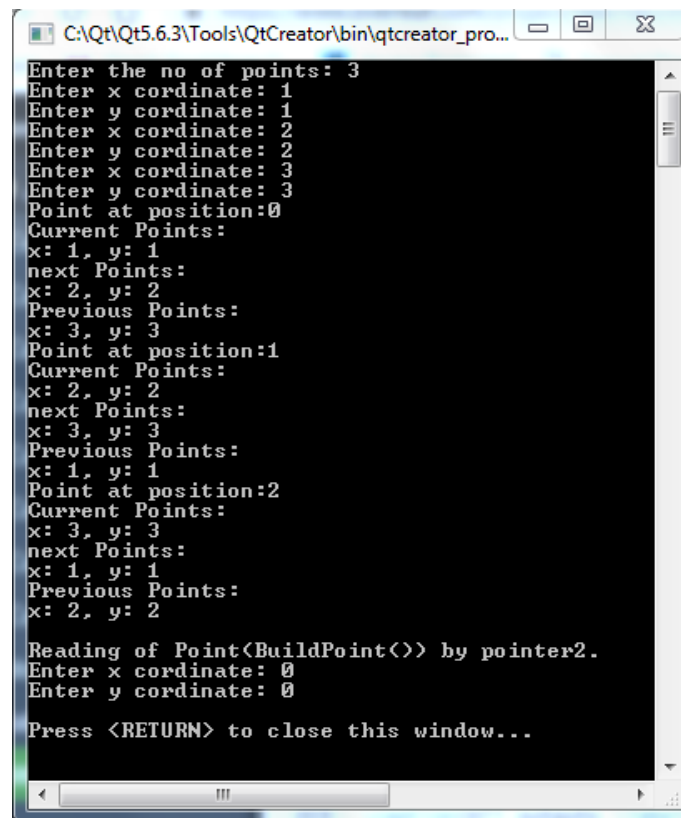
Point at position:0

Current Points:
x: 1, y: 1
next Points:
x: 2, y: 2
Previous Points:
x: 3, y: 3
Point at position:1

Current Points:
x: 2, y: 2
next Points:
x: 3, y: 3
Previous Points:
x: 1, y: 1
Point at position:2

Current Points:
x: 3, y: 3
next Points:
x: 1, y: 1
Previous Points:
x: 2, y: 2
Press <RETURN> to close this window...
```

Figure 2: All outputs

A screenshot of a Qt Creator console window. The title bar shows the path 'C:\Qt\Qt5.6.3\Tools\QtCreator\bin\qtcreator_pro...'. The console output is as follows:

```
Enter the no of points: 3
Enter x cordinate: 1
Enter y cordinate: 1
Enter x cordinate: 2
Enter y cordinate: 2
Enter x cordinate: 3
Enter y cordinate: 3
Point at position:0
Current Points:
x: 1, y: 1
next Points:
x: 2, y: 2
Previous Points:
x: 3, y: 3
Point at position:1
Current Points:
x: 2, y: 2
next Points:
x: 3, y: 3
Previous Points:
x: 1, y: 1
Point at position:2
Current Points:
x: 3, y: 3
next Points:
x: 1, y: 1
Previous Points:
x: 2, y: 2

Reading of Point<BuildPoint(>> by pointer2.
Enter x cordinate: 0
Enter y cordinate: 0

Press <RETURN> to close this window...
```

Figure 3: All outputs