National University of Computer and Emerging Sciences



Lab Exercise 13

For

Object Oriented Programming Lab

|  |  |
| --- | --- |
| Name | Muhammad Zain |
| Roll No | 19F-0228 |
| Course Instructor(s) | Dr. Danish |
| Lab Instructor(s) | Mr. Mughees Ismail |
| Semester | Spring 2020 |

|  |
| --- |
| **Question#1** |

☹,☹

|  |
| --- |
| **Question#2** |

# Source Code:

#include<iostream>

#include<math.h>

using namespace std;

class Polynomial

{

private:

int Array[100];

int x;

public:

Polynomial() {}

//simple constructor

void getData(int size)

{

for (int i = 0; i <= size; i++)

{

cout << "Enter the " << i + 1 << " Value : " ;

cin >> Array[i];

}cout << endl;

}

void disp(int size)

{

int dummy;

dummy = size;

for (int i = 0; i <= size; i++)

{

if (i == 0)

{

cout << Array[i] << "x^" << size;

}

else if (i < size && i>0)

{

cout << " + " << Array[i] << "x^" << --dummy;

}

else if (i == size)

{

cout << " + " << Array[i];

}

}

cout << endl;

}

Polynomial operator +(Polynomial obj\_second)

{

Polynomial t;

for (int i = 0; i <= 3; i++)

{

t.Array[i] = Array[i] + obj\_second.Array[i];

}

return t;

}

Polynomial operator +=(Polynomial obj\_second)

{

Polynomial t;

for (int i = 0; i <= 3; i++)

{

t.Array[i] = Array[i] + obj\_second.Array[i];

}

return t;

}

Polynomial operator -(Polynomial obj\_second)

{

Polynomial t;

for (int i = 0; i <= 3; i++)

{

t.Array[i] = Array[i] - obj\_second.Array[i];

}

return t;

}

Polynomial operator -=(Polynomial obj\_second)

{

Polynomial t;

for (int i = 0; i <= 3; i++)

{

t.Array[i] = Array[i] - obj\_second.Array[i];

}

return t;

}

Polynomial operator \*(Polynomial obj\_second)

{

Polynomial t;

for (int i = 0; i <= 3; i++)

{

t.Array[i] = Array[i] \* obj\_second.Array[i];

}

return t;

}

Polynomial operator \*=(Polynomial obj\_second)

{

Polynomial t;

for (int i = 0; i <= 3; i++)

{

t.Array[i] = Array[i] \* obj\_second.Array[i];

}

return t;

}

Polynomial operator ==(Polynomial obj\_second)

{

Polynomial t;

for (int i = 0; i <= 3; i++)

{

t.Array[i] = Array[i];

Array[i]=obj\_second.Array[i];

obj\_second.Array[i] = t.Array[i];

}

return t;

}

void out\_for\_multiplication(int size)

{

int dummy;

dummy = size;

for (int i = 0; i <= size; i++)

{

if (i == 0)

{

cout << Array[i] << "x^" << dummy+dummy;

}

else if (i < size && i>0)

{

int temp;

temp = --dummy;

cout << " + " << Array[i] << "x^" << temp+temp;

}

else if (i == size)

{

cout << " + " << Array[i];

}

}

cout << endl;

}

};

int main()

{

int y;

cout << "Enter the size of Array" << endl;

cin >> y;

Polynomial obj, obj2;

cout << "Enter the Values of First Array" << endl;

obj.getData(y);

cout << "Enter the Values of Second Array" << endl;

obj2.getData(y);

cout << endl << endl;

cout << "Expression of 1st Polynomial : ";

obj.disp(y);

cout << "Expression of 2nd Polynomial : ";

obj2.disp(y);

cout << endl << endl;

Polynomial obj3;//for storage of addition

obj3 = obj2 + obj;

cout << "Sum with + = ";

obj3.disp(y);

Polynomial obj4;//for storage of subtraction

obj4 = obj2 - obj;

cout << "Difference with - = ";

obj4.disp(y);

cout << endl;

Polynomial obj5;//for storage of addition

obj5 = obj2 += obj;

cout << "Sum with -= = ";

obj5.disp(y);

cout << endl;

Polynomial obj6;//for storage of subtraction

obj6 = obj2 -= obj;

cout << "Difference with -= = ";

obj4.disp(y);

cout << endl;

Polynomial obj7;//for storage of Multiplication

obj7 = obj2 \* obj;

cout << "Multiplication with \* = ";

obj7.out\_for\_multiplication(y);

cout << endl;

Polynomial obj8;//for storage of Multiplication

obj8 = obj2 \*= obj;

cout << "Multiplication with \*= = ";

obj8.out\_for\_multiplication(y);

cout << endl;

Polynomial obj9;//for storage of subtraction

obj9 = obj2 == obj;

cout << "Assinging 1 to 2 with -= = ";

obj9.disp(y);

cout << endl;

system("Pause>0");

}

# Snip:

# 

|  |
| --- |
| **Question#3** |

# Source Code:

#include<iostream>

#include<string>

using namespace std;

/\*

(Date Class) Defining Operator (+=) and

Operator (-=) to Add or Subtract Days in the

object Given an Integer Input.

\*/

#include<iostream>

using namespace std;

class Day\_Change {

protected:

int day;

int month;

int year;

public:

Day\_Change() {}

Day\_Change(int x, int y, int z)

{

day = x;

month = y;

year = z;

}

friend Day\_Change operator +=(Day\_Change&, int);

friend Day\_Change operator -=(Day\_Change&, int);

void output()

{

cout << "Date : " << day << " - " << month << " - " << year << endl;

}

};

Day\_Change operator +=(Day\_Change&obj, int input)//jo user input kray ga k itna plus krna

{

return Day\_Change(obj.day + input, obj.month, obj.year);

}

Day\_Change operator -=(Day\_Change&obj, int input)//jo user input kray ga k itna plus krna

{

return Day\_Change(obj.day - input, obj.month, obj.year);

}

int main()

{

int Day, Month, Year, plus, minus;

cout << "Input Day " << endl;

cin >> Day;

cout << "Input Month" << endl;

cin >> Month;

cout << "Input Year " << endl;

cin >> Year;

cout << endl << endl;

cout << "Enter number of Days to Add " << endl;

cin >> plus;

Day\_Change Obj\_1(Day, Month, Year);//parametrized constructor

Day\_Change Obj\_2 = Obj\_1 += plus;

Obj\_2.output();

cout << endl << endl;

cout << "Enter number of Days to Subtract " << endl;

cin >> minus;

Day\_Change Obj\_3 = Obj\_2 -= minus;//in d2

Obj\_3.output();

system("pause>0");

}

# Snip:

