NAME: Satvik Dandale

Roll No: 13

GR No: 1710797

DIV: TY-E

BATCH: B1

/\*

Write a CPP program to implement following polynomial operations:

1. Addition

2. Subtraction

3. Multiplication

4. Evaluation

using OOP concept.

\*/

#include<bits/stdc++.h>

using namespace std;

class Polynomial{

private:

float \*coefficients;

// This array will store the coefficients of all the terms in the polynomial.

// The length of this array will be the degree of polynomial + 1;

char variable;

// This will store the variable of the polynomial.

int degree;

// Highest degree of the polynomial

public:

// Constructors

Polynomial(){

this->coefficients = NULL;

this->degree = -1; // No polynomial assigned yet

this->variable = 'x';

}

Polynomial(int degree){

this->degree = degree;

coefficients = new float[degree + 1];

for(int i=0; i < degree + 1; i++)

this->coefficients[i] = 0;

}

// Copy Constructor

Polynomial(Polynomial &p1){

// Initialising the array

this->degree = p1.degree;

this->coefficients = new float [degree + 1];

for(int i=0; i < degree + 1; i++)

this->coefficients[i] = p1.coefficients[i]; // Copying each element

// Copying the variable

this->variable = p1.variable;

}

void newPolynomial(){

// This function will provide an interface for the user to initialize the new polynomial.

cout<<"Enter the degree of the polynomial (Highest Degree):\n";

cin>>this->degree;

this->coefficients = new float[this->degree + 1];

for(int i=0; i < degree + 1; i++){

cout<<"Enter the coefficient for "<<variable<<"^"<<i<<endl;

cin>>this->coefficients[i];

}

}

void display(){

if (this->degree == -1)

return;

// Display the polynomial in the decreasing order of the degree.

for(int i = this->degree; i >= 0; i--){

if (this->coefficients[i] < 0 || i == this->degree){

// Don't put "+" sign for the first term or a negative term.

if (i == 0) // Don't show the variable

cout<<this->coefficients[i]<<this->variable<<" ";

else

cout<<this->coefficients[i]<<this->variable<<"^"<<i<<" ";

}

else if(this->coefficients[i] == 0){

// Don't show this term.

// This term doesn't exist.

continue;

}

else{

if (i == 0) // Don't show the variable

cout<<"+ "<<this->coefficients[i]<<" ";

else

cout<<"+ "<<this->coefficients[i]<<this->variable<<"^"<<i<<" ";

}

}

}

// Arithmatic Operations

Polynomial\* add(Polynomial &p){

// There are two polynomials:

// this and p

// Create a new Polynomial Object

Polynomial \*temp;

// Initialize the temp Polynomial object with the char and the greater degree

//Check which is longer:

if (this->degree > p.degree){

temp = new Polynomial(this->degree);

// Loop within "this"

for(int i=0; i < this->degree + 1; i++){

if (i < p.degree + 1){

temp->coefficients[i] = this->coefficients[i] + p.coefficients[i];

}// Else the second array has been exhausted.

else

temp->coefficients[i] = this->coefficients[i];

}// ADDING IS DONE

}

else{

temp = new Polynomial(p.degree);

// Loop withing "p"

for(int i=0; i < p.degree + 1; i++){

if (i < this->degree + 1){

temp->coefficients[i] = p.coefficients[i] + this->coefficients[i];

}// Else the second array has been exhausted

else

temp->coefficients[i] = p.coefficients[i];

}// ADDING IS DONE

}

return temp;

}// Function ends

Polynomial\* sub(Polynomial &p){

// There are two polynomials:

// this and p

// Create a new Polynomial Object

Polynomial \*temp;

// Initialize the temp Polynomial object with the char and the greater degree

//Check which is longer:

if (this->degree > p.degree){

temp = new Polynomial(this->degree);

// Loop within "this"

for(int i=0; i < this->degree + 1; i++){

if (i < p.degree + 1){

temp->coefficients[i] = this->coefficients[i] - p.coefficients[i];

}// Else the second array has been exhausted.

else

temp->coefficients[i] = this->coefficients[i];

}// ADDING IS DONE

}

else{

temp = new Polynomial(p.degree);

// Loop withing "p"

for(int i=0; i < p.degree + 1; i++){

if (i < this->degree + 1){

temp->coefficients[i] = this->coefficients[i] - p.coefficients[i];

}// Else the second array has been exhausted

else

temp->coefficients[i] = p.coefficients[i];

}// ADDING IS DONE

}

return temp;

}// Function ends

Polynomial\* multiply(Polynomial &p){

// Two polys

// "this" and "p"

// The highest degree of the product will be (degree of "this") + (degree of "p")

Polynomial \*temp = new Polynomial(this->degree + p.degree);

// For every term in "this", it is multiplied with all the terms of the other polynomial.

for(int i=0; i < this->degree + 1; i++){

for(int j=0; j < p.degree + 1; j++){

int degree = i + j; // Degrees will be added

int coff = this->coefficients[i] \* p.coefficients[j];

temp->coefficients[degree] += coff;

}

}

return temp;

}//Function ends

// Overloaded Operators

Polynomial \*operator+(Polynomial &p){

return this->add(p);

}

Polynomial \*operator-(Polynomial &p){

return this->sub(p);

}

Polynomial \*operator\*(Polynomial &p){

return this->multiply(p);

}

float evaluate(float value){

// This will evaluate the polynomial by placing the value of x as the given value

float result = 0;

for(int i=0; i < this->degree + 1; i++)

result += this->coefficients[i] \* pow(value, i);

return result;

}

};

int main(){

Polynomial \*p = new Polynomial();

p->newPolynomial();

Polynomial \*p1 = new Polynomial();

p1->newPolynomial();

cout<<"First Polynomial is:\n";

p->display();

cout<<"\nSecond Polynomial is:\n";

p1->display();

cout<<"\nAddition:\n";

Polynomial \*result = new Polynomial(\*(\*p + \*p1));

result->display();

cout<<"\nSubtraction:\n";

result = new Polynomial(\*(\*p - \*p1));

result->display();

cout<<"\nMultiplication:\n";

result = new Polynomial(\*(\*p \* \*p1));

result->display();

cout<<"\nEnter the value of x for the polynomial: ";

p->display();

cout<<endl;

int value;

cin>>value;

cout<<"The answer after evaluating this polynomial using the value of x as "<<value;

cout<<" is\n"<<p->evaluate(value)<<endl;

}

OUTPUT:

Enter the degree of the polynomial (Highest Degree):

3

Enter the coefficient for x^0

9

Enter the coefficient for x^1

8

Enter the coefficient for x^2

6

Enter the coefficient for x^3

4

Enter the degree of the polynomial (Highest Degree):

2

Enter the coefficient for x^0

1

Enter the coefficient for x^1

5

Enter the coefficient for x^2

9

First Polynomial is:

4x^3 + 6x^2 + 8x^1 + 9

Second Polynomial is:

9x^2 + 5x^1 + 1

Addition:

4^3 + 15^2 + 13^1 + 10

Subtraction:

4^3 -3^2 + 3^1 + 8

Multiplication:

36^5 + 74^4 + 106^3 + 127^2 + 53^1 + 9

Enter the value of x for the polynomial: 4x^3 + 6x^2 + 8x^1 + 9

3

The answer after evaluating this polynomial using the value of x as 3 is

195