Assignment No. 02

Roll No. SC55 - Shreyas Chavhan

===================================================

# Quadratic Class

## Problem Statement -

Implement a class Quadratic that represents two degree polynomials i.e.,

polynomials of type ax2+bx+c. Your class will require three data member corresponding

to a, b and c. Implement the following operations:

1.A constructor(including a default constructor which creates the 0 polynomial).

2.Overloaded operator+ to add two polynomials of degree 2.

3.Overloaded << and >> to print and read polynomials.

To do this, you will need to decide what you want your input and output format to look like.

4.A function eval that computes the value of a polynomial for a given value of x.

5.A function that computes the two solutions of the equation ax 2 +bx+c=0

## Outcomes -

To understand operator overloading concept.

## Theory -

### Operator Overloading:

You can redefine or overload most of the built-in operators available in

C++. Thus a programmer can use operators with user-defined types as well.

Overloaded operators are functions with special names the keyword operator followed by the

symbol for the operator being defined. Like any other function, an overloaded operator has a

return type and a parameter list.

Box operator+(const Box&amp;);

declares the addition operator that can be used to add two Box objects and returns final Box

object.

Roll No. SC55 - Shreyas Chavhan

===================================================

### Overloadable/Non-overloadableOperators:

Following is the list of operators which can be overloaded:

+ - \* / % ^

&amp; | ~ ! , =

+= -= /= %= ^= &amp;=

|= \*= &lt;&lt;= &gt;&gt;= [ ] ( )

-&gt; -&gt;\* new new [ ] delete delete [ ]

Operator that are not overloaded are follows

scope operator - ::

sizeof

member selector - .

member pointer selector - \*

ternary operator - ?:

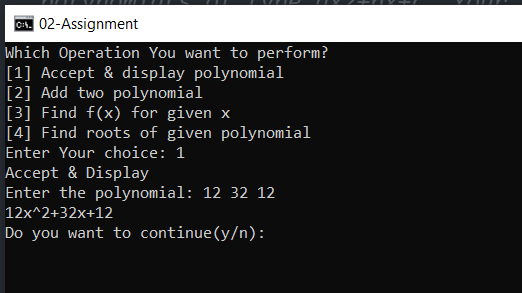
## Code -

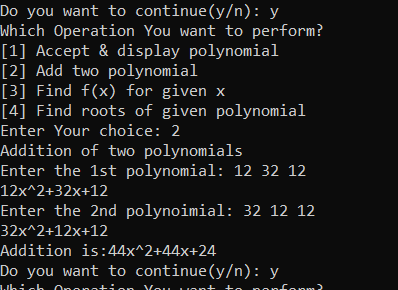
|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Implement a class Quadratic that represents two degree polynomials i.e.,  polynomials of type ax2+bx+c. Your class will require three data member corresponding  to a, b and c. Implement the following operations:  1.A constructor(including a default constructor which creates the 0 polynomial).  2.Overloaded operator+ to add two polynomials of degree 2.  3.Overloaded << and >> to print and read polynomials.  To do this, you will need to decide what you want your input and output format to look like.  4.A function eval that computes the value of a polynomial for a given value of x.  5.A function that computes the two solutions of the equation ax 2 +bx+c=0  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include<iostream>  #include<cmath>  using namespace std;  class Quadratic  {  int a,b,c;  public:  Quadratic() //default constructor  {  a = 0;  b = 0;  c = 0;  }  Quadratic(int x, int y, int z) //parameterized constructor  {  a = x; b = y; c = z;  }  friend istream &operator >> (istream &IN, Quadratic &T) //operator >> overloaded  {  IN >> T.a >> T.b >> T.c;  return IN;  }  friend ostream &operator << (ostream &OUT, Quadratic &T) //operator << overloaded  {  OUT << T.a << "x^2+" << T.b << "x+" << T.c;  return OUT;  }  Quadratic operator + (Quadratic T) //opeartor + overloaded  {  Quadratic R;  R.a = a + T.a;  R.b = b + T.b;  R.c = c + T.c;  return R;  }  void eval(Quadratic T,int x) //eval fn to evaluate the polynomial for given value of x  {  int z;  z = T.a \* x \* x + T.b \* x + T.c;  cout << "for x = " << x << ", f(" << x << ") = " << z << "\n";  }  void compute(Quadratic T) //compute fn to find roots of polynomial  {  float x, y1, y2;  x = T.b \* T.b-4 \* T.a \* T.c;  if(x > 0)  {  cout << "Roots are real & not equal\n";  y1 = (-T.b+sqrt(x)) / (2 \* T.a);  y2 = (-T.b-sqrt(x)) / (2 \* T.a);  cout << y1 << "\n";  cout << y2 << "\n";  }  else if(x == 0)  {  cout << "Roots are real & equal\n";  y1 = -T.a / (2\*T.a);  cout << y1 << "\n";  }  else if(x < 0)  {  cout << "complex Roots\n";  }  }  };  int main()  {  int x, ch;  char p;  Quadratic s1(5,6,10), s2, s3;  do  {  cout << "Which Operation You want to perform?\n";  cout << "[1] Accept & display polynomial\n";  cout << "[2] Add two polynomial\n";  cout << "[3] Find f(x) for given x\n";  cout << "[4] Find roots of given polynomial\n";  cout << "Enter Your choice: ";  cin >> ch;  switch(ch)  {  case 1:  cout << "Accept & Display\n";  cout << "Enter the polynomial: ";  cin >> s2;  cout << s2 << endl;  break;  case 2:  cout << "Addition of two polynomials\n";  cout << "Enter the 1st polynomial: ";  cin >> s1;  cout << s1 << endl;  cout << "Enter the 2nd polynoimial: ";  cin >> s2;  cout << s2 << endl;  s3 = s1 + s2;  cout << "Addition is:";  cout << s3 << endl;  break;  case 3:  cout << "Calculate f(X) for given x\n";  cout << "Enter the polynomial: ";  cin >> s1;  cout << "Enter the value for x = ";  cin >> x;  s2.eval(s1,x);  break;  case 4:  cout << "find roots of polynomial\n";  cout << "enter the polynomial: ";  cin >> s2;  s2.compute(s2);  break;  }  cout << "Do you want to continue(y/n): ";  cin >> p;  }while( p == 'y' || p == 'Y');  return 0;  } |

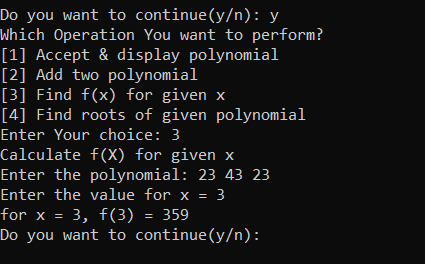
Roll No. SC55 - Shreyas Chavhan

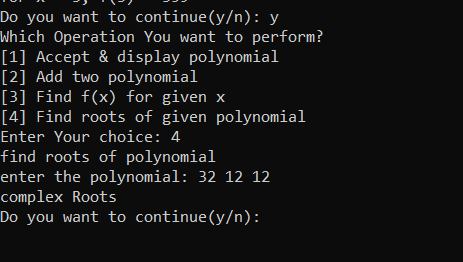
===================================================

## Output -









===================================================

Roll No. SC55 - Shreyas Chavhan