Shaurya Singh Srinet  
 RA2111032010006   
 T2   
 CSE IoT

Week-6

**Q1**. Write a C++ Program to Compare Two Strings using binary operator overloading.

# CODE:

#include <cstring> #include <iostream> #include <string.h> using namespace std; class CompareString { public:

char str[25]; CompareString(char str1[])

{

strcpy(this->str, str1);

}

int operator==(CompareString s2)

{

if (strcmp(str, s2.str) == 0) return 1;

else

return 0;

}

int operator<=(CompareString s3)

{

if (strlen(str) <= strlen(s3.str)) return 1;

else

return 0;

}

int operator>=(CompareString s3)

{

if (strlen(str) >= strlen(s3.str)) return 1;

else

return 0;

}

};

void compare(CompareString s1, CompareString s2)

{

if (s1 == s2)

cout << s1.str << " is equal to "<< s2.str << endl; else {

cout << s1.str << " is not equal to "<< s2.str << endl; if (s1 >= s2)

cout << s1.str << " is greater than "<< s2.str << endl; else

cout << s2.str << " is greater than "<< s1.str << endl;

}

}

void testcase1()

{

char str1[] = "Geeks"; char str2[] = "ForGeeks"; CompareString s1(str1); CompareString s2(str2);

cout << "Comparing \"" << s1.str << "\" and \""<< s2.str << "\"" << endl; compare(s1, s2);

}

void testcase2()

{

char str1[] = "Geeks"; char str2[] = "Geeks"; CompareString s1(str1); CompareString s2(str2);

cout << "\n\nComparing \"" << s1.str << "\" and \""<< s2.str << "\"" << endl; compare(s1, s2);

}

int main()

{

testcase1(); testcase2();

return 0;

}

# OUTPUT:

Geeks is not equal to ForGeeks ForGeeks is greater than Geeks Geeks is equal to Geeks

**Q2.** Write a C++ Program to Perform Complex Operations using binary operator overloading.

**CODE:**

#include <iostream> using namespace std; class Complex\_num

{

int x, y; public:

void inp()

{

cout << " Input two complex number: " << endl; cin >> x >> y;

}

Complex\_num operator + (Complex\_num obj)

{

Complex\_num A;

A.x = x + obj.x;

A.y = y + obj.y; return (A);

}

Complex\_num operator - (Complex\_num obj)

{

Complex\_num A;

A.x = x - obj.x;

A.y = y - obj.y; return (A);

}

void print()

{

cout << x << " + " << y << "i" << "\n";

}

void print2()

{

cout << x << " - " << y << "i" << "\n";

}

};

int main ()

{

Complex\_num x1, y1, sum, sub; x1.inp();

y1.inp();

sum = x1 + y1; sub = x1 - y1;

cout << "\n Entered values are: \n"; cout << " \t";

x1.print(); cout << " \t"; y1.print();

cout << "\n The addition of two complex (real and imaginary) numbers: "; sum.print();

cout << "\n The subtraction of two complex (real and imaginary) numbers: "; sub.print2();

return 0;

}

# OUTPUT:

Input two complex numbers: 5

7

Input two complex numbers: 3

5

Entered values are: 5 + 7i

3 + 5i

The addition of two complex (real and imaginary) numbers: 8 + 12i The subtraction of two complex (real and imaginary) numbers: 2 - 2i

**Q3.** Write a C++ program to Swap variables using binary operator overloading.

# CODE:

#include<bits/stdc++.h> using namespace std; int main()

{

int a = 7, b = 3;

cout << "Before swaping a = " << a << " , b = " << b << endl; a = a ^ b;

b = b ^ a; a = a ^ b;

cout << "After swaping a = " << a << " , b = " << b << endl; return 0;

}

# OUTPUT:

Before swaping a = 7 , b = 3 After swaping a = 3 , b = 7

**Q4.** Write a C++ program for show Counter using Overloading unary operator ++.

# CODE:

#include<iostream> using namespace std; class counter{

int count; public: counter(){

count=0;

}

int get\_count(){ return count;

}

void operator++(){ count++;

}

};

int main(){ counter c1,c2;

cout<<"\nC1 ="<<c1.get\_count(); cout<<"\nC2 ="<<c2.get\_count();

c1++; c2++;

++c2;

cout<<"\nC1 ="<<c1.get\_count(); cout<<"\nC2 ="<<c2.get\_count(); return 0;

}

# OUTPUT:

C1=0 C2=O C1=1 C2=2

**Q5**. Write a C++ class Program to perform rational number arithmetic using operator overloading.

# CODE:

#include<stdio.h> #include<iostream> using namespace std; class rational{

int numer; int denom; public:

void getdata(){

cout<<"\n Enter the numerator part of the rational no. : "; cin>>numer;

cout<<"\n Enter the denominator part of the rational no. : "; cin>>denom;

}

void operator+(rational); void operator-(rational);

};

void rational ::operator+(rational c1){ rational temp;

temp.numer=(numer\*c1.denom)+(c1.numer\*denom); temp.denom=denom\*c1.denom;

cout<<"\nRational no. after addition : ";

cout<<"\n Numerator = "<<temp.numer<<"\n Denominator = "<<temp.denom;

}

void rational ::operator -(rational c1){ rational temp;

temp.numer=(numer\*c1.denom)-(c1.numer\*denom); temp.denom=denom\*c1.denom;

cout<<"\n Rational no. after subtraction : ";

cout<<"\n Numerator = " <<temp.numer<<"\n Denominator = "<<temp.denom;

}

int main(){

rational c1, c2; int n;

cout<<endl<<"\n Enter the data for first rational no.: "; c1.getdata();

cout<<endl<<"\n Enter the data for second rational no. : "; c2.getdata ();

c1+c2; c1-c2; return 0;

}

# OUTPUT:

Enter the data for first rational no.:

Enter the numerator part of the rational no.: 3 Enter the denominator part of the rational no. : 7

Enter the data for second rational no. :

Enter the numerator part of the rational no.: 2 Enter the denominator part of the rational no. : 9

Rational no. after addition :

Numerator = 41

Denominator = 63

Rational no. after subtraction :

Numerator = 13

Denominator = 63

**Q6.** Write a C++ class Program to perform Complex Arithmetic using operator overloading

# CODE:

#include<iostream> #include<stdio.h> using namespace std; class complex{

int real; float image; public:

void getdata(){

cout<<"\n enter the real part of the complex :: "; cin>>real;

cout<<"\n enter the imaginary part of the complex :: "; cin>>image;

}

void operator + (complex); void operator - (complex);

};

void complex :: operator + (complex c1){ complex temp; temp.real=real+c1.real;

temp.image=image+c1.image; if (temp.image>=0){

cout<<"\n complex no. after addition :: "; cout<<temp.real<<"+"<<temp.image<<"i\n";

}

else{

cout<<"\n complex no. after addition :: "; cout<<temp.real<<temp.image<<"i\n";

}

}

void complex ::operator-(complex c1){ complex temp;

temp.real = real-c1.image; temp.image= image-c1.image; if (temp.image>=0){

cout<<"\n complex no. after subtraction :: "; cout<<"\n temp.real"<<"+"<<temp.image<<"i\n";

}

else{

cout<<"\n complex no. after subtraction :: "; cout<<temp.real<<temp.image<<"i\n";

}

}

int main(){

complex c1, c2; int n;

cout<<endl<<"\n Enter the data for First Complex No. ";

c1.getdata();

cout<<endl<<"\n Enter the data for seconds Complex No. ";

c2.getdata(); c1+c2;

c1-c2; return 0;

}

# OUTPUT:

Enter the data for First Complex No......

enter the real part of the complex :: 2

enter the imaginary part of the complex :: 3

Enter the data for seconds Complex No.....

enter the real part of the complex :: 4

enter the imaginary part of the complex :: 5

complex no. after addition :: 6+8i complex no. after subtraction :: -3-2i

**Q7.** Write a C++ Program to Find the Area of shapes using function and operator overloading.

# CODE:

#include<iostream> using namespace std; int area(int);

int area(int,int); float area(float);

float area(float,float); int main(){

int s,l,b; float r,bs,ht;

cout<<"Enter side of a square:"; cin>>s;

cout<<"Enter length and breadth of rectangle:"; cin>>l>>b;

cout<<"Enter radius of circle:"; cin>>r;

cout<<"Enter base and height of triangle:"; cin>>bs>>ht;

cout<<"Area of square is"<<area(s); cout<<"\nArea of rectangle is "<<area(l,b); cout<<"\nArea of circle is "<<area(r); cout<<"\nArea of triangle is "<<area(bs,ht);

}

int area(int s){ return(s\*s);

}

int area(int l,int b){ return(l\*b);

}

float area(float r){ return(3.14\*r\*r);

}

float area(float bs,float ht){ return((bs\*ht)/2);

}

# OUTPUT:

Enter side of a square:2

Enter length and breadth of rectangle:3 6 Enter radius of circle:3

Enter base and height of triangle:4 4

Area of square is 4 Area of rectangle is 18 Area of circle is 28.26 Area of triangle is 8

**Q8.** Write a C++ Program to check Palindrome using function and operator overloading

# CODE:

#include<iostream> #include<string.h> using namespace std; class func

{

public:

void pal(int ); void pal(char c[]);

};

void func :: pal(int n){ int n1,sum,a; n1=n;

sum=0; do

{

sum=sum\*10; a=n%10;

sum=sum+a; n=n/10;

}

while(n>0); if(sum==n1)

cout<<"\n It is Palindrome";

else

}

cout<<"\n It is not a palindrome";

void func :: pal(char c[])

{

int l=strlen(c); char b[10]; int i,j;

for(i=0,j=l-1;i<l;i++,j--)

{

b[i]=c[j];

}

b[i]=' ';

if(strcmp(c,b)==0)

{cout<<"\nThe entered text is palindrome";

}

else

{cout<<"\n The entered text is not a palindrome";

}

}

int main(){

char c[10]; int k;

func p;

cout<<"Enter the number: ";

cin>>k; p.pal(k);

cout<<"\nEnter the string to be checked:"<<endl; cin>>c;

p.pal(c); return 0;

}

# OUTPUT:

Enter the number: 1234321 It is Palindrome

Enter the string to be checked: asdfgh The entered text is not a palindrome

**Q9.** Write a C++ program to find area of square, rectangle, Circle and triangle using operator overloading.

# CODE:

#include<iostream> using namespace std; int area(int);

int area(int,int); float area(float);

float area(float,float); int main(){

int s,l,b; float r,bs,ht;

cout<<"Enter side of a square:"; cin>>s;

cout<<"Enter length and breadth of rectangle:"; cin>>l>>b;

cout<<"Enter radius of circle:"; cin>>r;

cout<<"Enter base and height of triangle:"; cin>>bs>>ht;

cout<<"Area of square is"<<area(s); cout<<"\nArea of rectangle is "<<area(l,b); cout<<"\nArea of circle is "<<area(r); cout<<"\nArea of triangle is "<<area(bs,ht);

}

int area(int s){ return(s\*s);

}

int area(int l,int b){ return(l\*b);

}

float area(float r){ return(3.14\*r\*r);

}

float area(float bs,float ht){ return((bs\*ht)/2);

}

# OUTPUT:

Enter side of a square:2

Enter length and breadth of rectangle:3 6 Enter radius of circle:3

Enter base and height of triangle:4 4

Area of square is 4 Area of rectangle is 18 Area of circle is 28.26 Area of triangle is 8

**Q10.** Write a C++ program to find volume of cube, cylinder, and sphere by function and operator overloading.

# CODE:

#include<iostream> using namespace std; float vol(int,int);

float vol(float); int vol(int);

int main(){ int r,h,a; float r1;

cout<<"Enter radius and height of a cylinder:"; cin>>r>>h;

cout<<"Enter side of cube:"; cin>>a;

cout<<"Enter radius of sphere: "; cin>>r1;

cout<<"Volume of cylinder is "<<vol(r,h); cout<<"\nVolume of cube is "<<vol(a); cout<<"\nVolume of sphere is "<<vol(r1); return 0;

}

float vol(int r,int h){ return(3.14\*r\*r\*h);

}

float vol(float r1){ return((4\*3.14\*r1\*r1\*r1)/3);

}

int vol(int a){ return(a\*a\*a);

}}

# OUTPUT:

Enter radius and height of a cylinder: 8 12 Enter side of cube: 2

Enter radius of sphere: 3

Volume of cylinder is 2411.52 Volume of cube is 8

Volume of sphere is 113.04