**LAB REPORT NO 7**



**Spring 2020**

**CSE102L Computer Programming Lab**

Submitted by:  **Muhammad Ali**

Registration No: **19PWCSE1801**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**MAM. Sumayyea salahuddin**

(January 12, 2021)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Activity 7.4.1:-**

**C++ code:-**

#include<iostream>

using namespace std;

class RationalNumber {

private:

int numerator, denomerator;

public:

RationalNumber() {

numerator = 0;

denomerator = 1;

}

RationalNumber(int a, int b) {

numerator = a;

denomerator = b;

}

void showRN() {

cout<<numerator <<"/"<<denomerator<<endl;

}

friend bool operator>=(RationalNumber, RationalNumber);

friend bool operator<=(RationalNumber, RationalNumber);

friend bool operator==(RationalNumber, RationalNumber);

friend RationalNumber operator+(RationalNumber, RationalNumber);

friend RationalNumber operator/(RationalNumber, RationalNumber);

friend bool operator>(RationalNumber, RationalNumber);

friend bool operator<(RationalNumber, RationalNumber);

friend RationalNumber operator-(RationalNumber, RationalNumber);

friend RationalNumber operator\*(RationalNumber, RationalNumber);

friend bool operator!=(RationalNumber, RationalNumber);

};

RationalNumber operator+(RationalNumber a, RationalNumber b) {

RationalNumber returnvalue(a.numerator \* b.denomerator + a.denomerator \* b.numerator,a.denomerator \* b.denomerator);

return returnvalue;

}

RationalNumber operator-(RationalNumber a, RationalNumber b) {

RationalNumber returnvalue(a.numerator \* b.denomerator - a.denomerator \* b.numerator,a.denomerator \* b.denomerator);

return returnvalue;

}

RationalNumber operator\*(RationalNumber a, RationalNumber b) {

RationalNumber returnvalue(a.numerator \* b.denomerator, a.denomerator \* b.denomerator);

return returnvalue;

}

RationalNumber operator/(RationalNumber a, RationalNumber b) {

RationalNumber returnvalue(a.numerator \* b.denomerator, a.denomerator \* b.numerator);

return returnvalue;

}

bool operator>(RationalNumber a, RationalNumber b) {

float n3 = a.numerator/a.denomerator, n4 = b.numerator/b.denomerator;

if(n3>n4) return true;

return false;

}

bool operator<(RationalNumber a, RationalNumber b) {

float n3 = a.numerator / a.denomerator, n4 = b.numerator/b.denomerator;

if(n3 < n4) return true;

return false;

}

bool operator<=(RationalNumber a, RationalNumber b) {

float n3 = a.numerator / a.denomerator, n4 = b.numerator/b.denomerator;

if(n3 <= n4) return true;

return false;

}

bool operator>=(RationalNumber a, RationalNumber b) {

float n3 = a.numerator / a.denomerator, n4 = b.numerator/b.denomerator;

if(n3 >= n4) return true;

return false;

}

bool operator==(RationalNumber a, RationalNumber b) {

float n3 = a.numerator / a.denomerator, n4 = b.numerator/b.denomerator;

if(n3 == n4) return true;

return false;

}

bool operator!=(RationalNumber a, RationalNumber b) {

float n3 = a.numerator / a.denomerator, n4 = b.numerator/b.denomerator;

if(n3!= n4) return true;

return false;

}

int main() {

RationalNumber a(2, 4), b(3, 9);

cout<<"A + B is ";

RationalNumber n = a + b;

n = a + b;

n.showRN();

a = a - b;

a.showRN();

a = a \* b;

a.showRN();

a = a / b;

a.showRN();

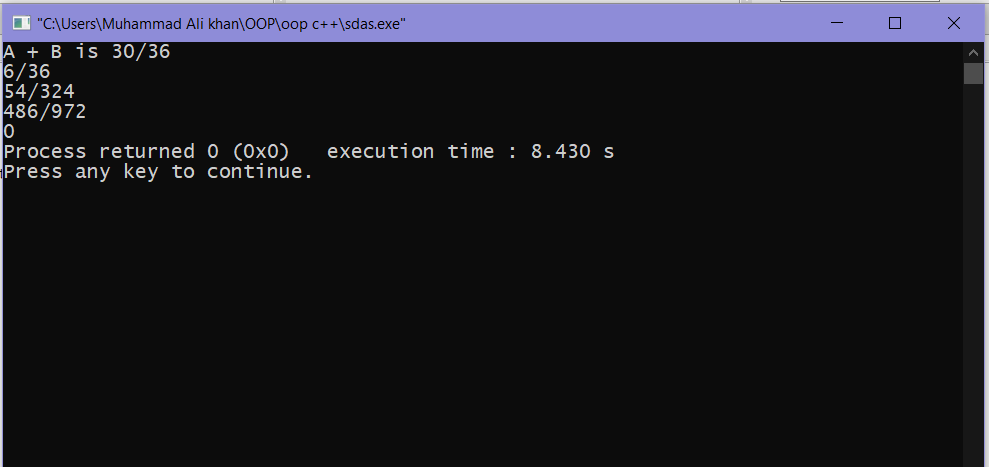
bool gr = a > b;

cout<<gr;

return 0;

}

**Output:-**

****

**Activity 7.4.1:-**

**C++ code:-**

#include<iostream>

using namespace std;

class Time {

private:

int s, m, h;

public:

Time(){

h = 0; m = 0; s = 0;

}

Time(int n1, int min, int sec) {

h = n1;

m = min;

s = sec;

}

void show() {

cout<<h<<":"<<m<<":"<<s<<endl;

}

friend Time operator+(Time, Time);

friend Time operator++(Time);

friend Time operator--(Time);

};

Time operator+(Time t1, Time t2) {

int hrs = t1.h + t2.h;

int min = t1.m + t2.m;

int sec = t1.s + t2.s;

if(sec > 59) {

sec = sec % 60;

min++;

}

if(min > 59) {

min = min % 60;

hrs++;

}

Time returnvalue(hrs, min, sec);

return returnvalue;

}

Time operator++(Time t) {

t.s++;

if(t.s > 59) {

t.m++;

t.s %= 60;

}

if(t.m > 59) {

t.h++;

t.m %= 60;

}

return t;

}

Time operator--(Time t) {

if(t.s == 0) {

if(t.m == 0) {

t.m = 59;

t.h--;

}else {

t.m--;

}

t.s = 59;

}else {

t.s--;

}

return t;

}

int main() {

Time t1(2, 1, 3), t2(1,1, 3);

cout<<"T1 is ";

t1.show();

cout<<"T2 is ";

t2.show();

cout<<"T1 + T2 is ";

Time t3 = t1 + t2;

t3.show();

t1 = ++t1;

t1.show();

t2 = --t2;

t2.show();

return 0;

}

**Output:-**

