OOP ASSIGNMENT#3

ROLL NO: 18F-0259
SECTION:C

NAME:RAZI ALLAH

Question No.1

Code

Header File(RationalNumber.h)

```
#pragma once
class RationalNumber
{
private:
	int numerator;
	int denominator;

public:
	RationalNumber();
	~RationalNumber();
	RationalNumber(int n, int d);
	RationalNumber operator+(RationalNumber a);

	void print();
	RationalNumber operator-(RationalNumber b);
	RationalNumber operator*(RationalNumber c);
	RationalNumber operator/(RationalNumber d);
};
```

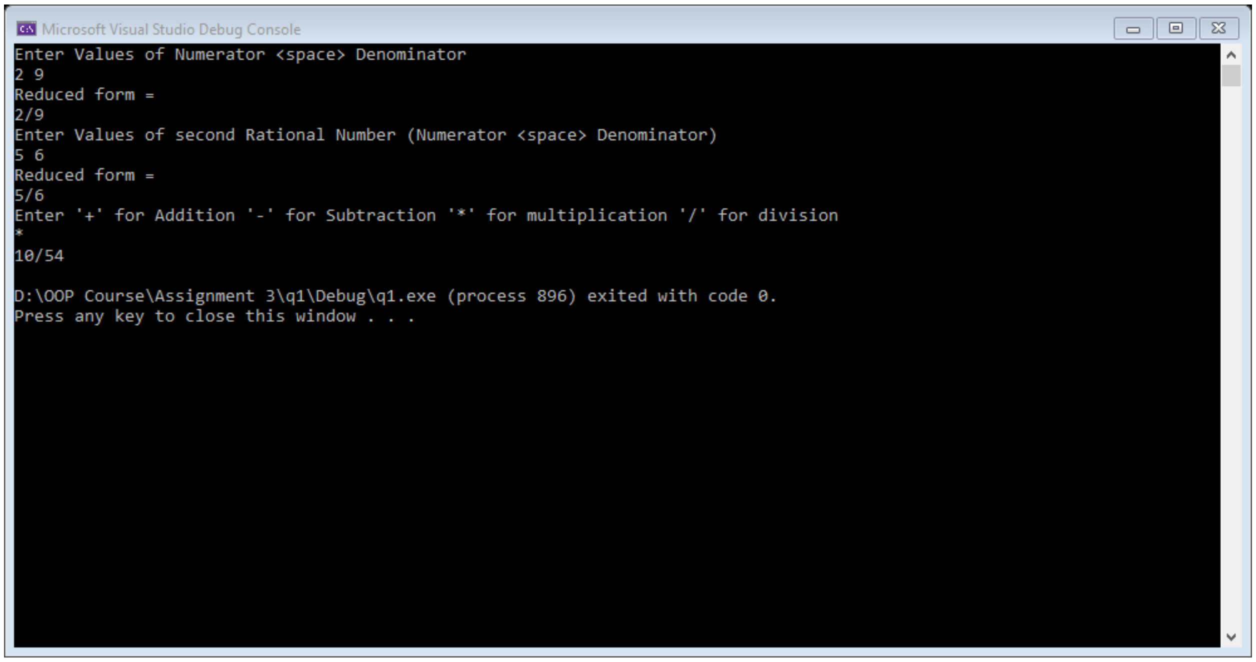
Cpp File

```
#include <iostream>
#include "RationalNumber.h"
using namespace std;

int main()
{
	char choice;
	int n = 0, d = 0;
	RationalNumber r1, r2, r3;
	cout << "Enter Values of Numerator <space> Denominator" << endl;
	cin >> n >> d;
	r1=RationalNumber(n, d);
	cout << "Enter Values of second Rational Number (Numerator <space> Denominator)" 
<< endl;
	cin >> n >> d;
	r2 = RationalNumber(n, d);
	cout << "Enter '+' for Addition '-' for Subtraction '*' for multiplication '/' for 
division" << endl;
	cin >> choice;
	if (choice=='+')
	{
		r3 = r2 + r1;
	}
	else if (choice=='-')
	{
```

```cpp
			r3 = r2 - r1;
		}
		else if (choice=='*')
		{
			r3 = r2 * r1;
		}
		else if (choice=='/')
		{
			r3 = r2 / r1;
		}
		r3.print();
}



RationalNumber::RationalNumber()
{
}


RationalNumber::~RationalNumber()
{
}

RationalNumber::RationalNumber(int n, int d)
{
	int max = 0;

	if (d==0||d<0)
	{
		cout << "Denominator can not be a zero or negative value" << endl;
	}
	if (n > d) max = n;
	else max = d;

	for (int i = 2; i <= max / 2; i++)
	{
		if (n % i == 0 && d % i == 0)
		{
			n /= i;
			d /= i;
		}
	}
	numerator = n;
	denominator = d;
	cout << "Reduced form =" << endl;
	cout << numerator << '/' << denominator << endl;
}
RationalNumber RationalNumber::operator+(RationalNumber a)
{
	int max;
	RationalNumber t;
	t.numerator = a.numerator * denominator + a.denominator * numerator;
	t.denominator = a.denominator * denominator;

	if (t.numerator > t.denominator) max = t.numerator;
    else max = t.denominator;
```

```
for (int i = 2; i <= max / 2; i++)
{
       if (t.numerator % i == 0 && t.denominator % i == 0)
       {
              t.numerator /= i;
              t.denominator /= i;
       }
}
       return t;
}
RationalNumber RationalNumber::operator-(RationalNumber b)
{
       RationalNumber t;
       t.numerator = b.denominator * numerator - denominator * b.numerator;
       t.denominator = b.denominator * denominator;
       return t;
}
RationalNumber RationalNumber::operator*(RationalNumber c)
{
       RationalNumber t;
       t.numerator = c.numerator * numerator;
       t.denominator = c.denominator * denominator;
       return t;
}
RationalNumber RationalNumber::operator/(RationalNumber d)
{
       RationalNumber t;
       t.numerator = d.denominator * numerator;
       t.denominator = denominator * d.numerator;
       return t;
}
void RationalNumber:: print()
{
       cout << numerator << '/' << denominator << endl;
}
```

Screen shot


Microsoft Visual Studio Debug Console

```
Enter Values of Numerator <space> Denominator
2 9
Reduced form =
2/9
Enter Values of second Rational Number (Numerator <space> Denominator)
5 6
Reduced form =
5/6
Enter '+' for Addition '-' for Subtraction '*' for multiplication '/' for division
*
10/54

D:\OOP Course\Assignment 3\q1\Debug\q1.exe (process 896) exited with code 0.
Press any key to close this window . . .
```

Question No.3

Code

Header File(Complex.h)

```
#pragma once
class Complex
{
private:
	int real;
	int imaginary;
public:
	Complex();
	~Complex();
	Complex(int r,int i);
	Complex operator+(Complex c1);
	Complex operator-(Complex c2);
	Complex operator *(Complex c3);
	Complex operator/(Complex c4);
	Complex operator==(Complex c5);
	Complex operator!=(Complex c6);
	void print();
};
```

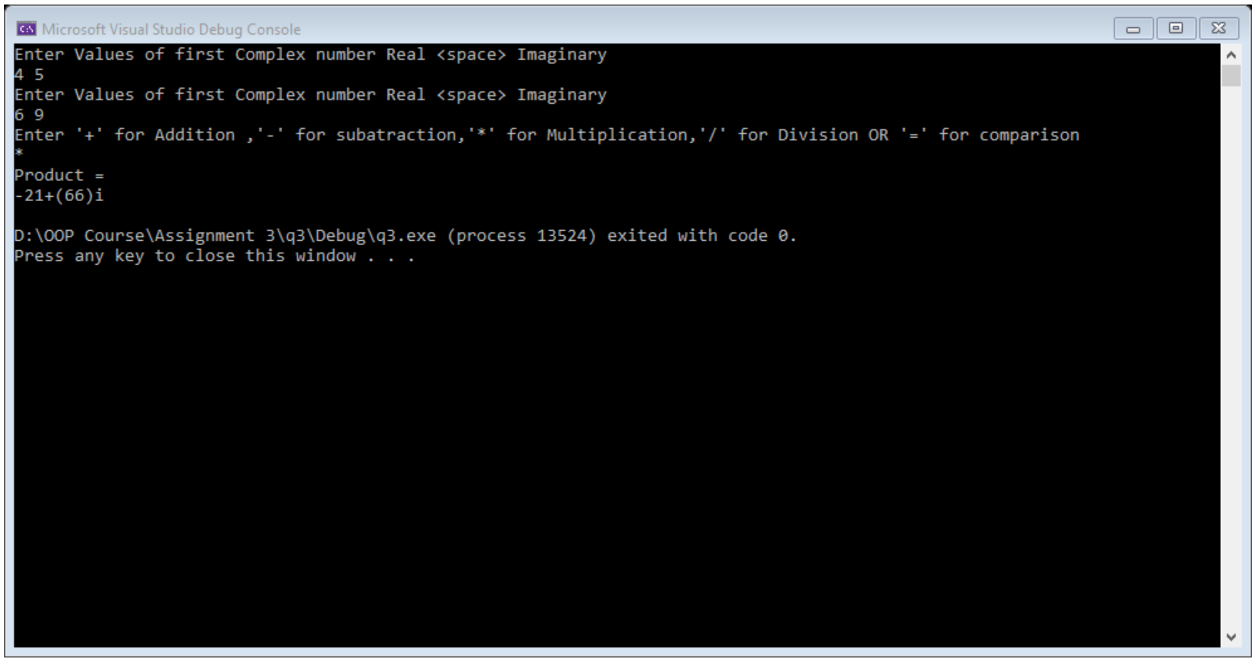
Cpp File

```
#include <iostream>
```

```
#include<math.h>
#include "Complex.h"
using namespace std;

int main()
{
	char choice;
	Complex c1,c2,c3;
	int real = 0, imaginary = 0;
	cout << "Enter Values of first Complex number Real <space> Imaginary" << endl;
	cin >> real >> imaginary;
	c1 = Complex(real, imaginary);
	cout << "Enter Values of first Complex number Real <space> Imaginary" << endl;
	cin >> real >> imaginary;
	c2 = Complex(real, imaginary);
	cout << "Enter '+' for Addition ,'-' for subatraction,'*' for Multiplication,'/'
for Division OR '=' for comparison" << endl;
	cin >> choice;
	switch (choice)
	{
	case'+':
		cout << "sum =" << endl;
		c3 = c2 + c1;
		c3.print();
		break;
	case'-':
		cout << "Diffrence =" << endl;
		c3 = c2 - c1;
		c3.print();
		break;
	case '*':
		cout << "Product =" << endl;
		c3 = c2 * c1;
		c3.print();
		break;
	case'/':
		cout << "Division =" << endl;
		c3 = c2 / c1;
		c3.print();
		break;
	case'=':
		c3 = c2 == c1;
		break;

	default:
		break;
	}
}

Complex::Complex()
{

}


Complex::~Complex()
{
}
```

```cpp
Complex::Complex(int r, int i)
{
	real = r;
	imaginary = i;
}
Complex Complex::operator+(Complex c1)
{
	Complex c;
	c.real = real + c1.real;
	c.imaginary = imaginary + c1.imaginary;
	return c;
}
Complex Complex::operator-(Complex c2)
{
	Complex c;
	c.real = real - c2.real;
	c.imaginary = imaginary - c2.imaginary;
	return c;
}
Complex Complex::operator*(Complex c3)
{
	Complex c;
	c.real = (real * c3.real) - (imaginary * c3.imaginary);
	c.imaginary = (real * c3.imaginary) + (c3.real * imaginary);
	return c;
}
Complex Complex::operator/(Complex c4)
{
	Complex c;
	c.real = (((real) * (c4.real)) + ((imaginary) * (c4.imaginary))) / (pow(c4.real,
2) + pow(c4.imaginary, 2));
	c.imaginary = (((c4.real) * (imaginary)) - ((real) * (c4.imaginary))) /
(pow(c4.real, 2) + pow(c4.imaginary, 2));
	return c;
}
Complex Complex::operator==(Complex c5)
{
	Complex c;
	if ((real = c5.real) && (imaginary = c5.imaginary))
	{
		cout << "two complex number entered are same" << endl;
	}
	else
	{
		c!= c5;
	}
	return c;
}
Complex Complex::operator!=(Complex c6)
{
	Complex c;
	cout << "two Complex numbers entered are not same" << endl;
	return c;
}
void Complex::print()
{
	cout << real << '+'<<'(' << imaginary <<')'<< 'i' << endl;
}
```

Screen Shot



```
Enter Values of first Complex number Real <space> Imaginary
4 5
Enter Values of first Complex number Real <space> Imaginary
6 9
Enter '+' for Addition ,'-' for subatraction,'*' for Multiplication,'/' for Division OR '=' for comparison
*
Product =
-21+(66)i

D:\OOP Course\Assignment 3\q3\Debug\q3.exe (process 13524) exited with code 0.
Press any key to close this window . . .
```

Question No.5

Code

Header File (CalenderDate.h)

```
#pragma once
class CalenderDate
{
private:
	int days;
	int months;
	int year;
public:
	CalenderDate();
	~CalenderDate();
	CalenderDate(int da, int mon, int ye);
	CalenderDate operator+=(CalenderDate a);
	CalenderDate operator-=(CalenderDate b);
	void print();
};
```

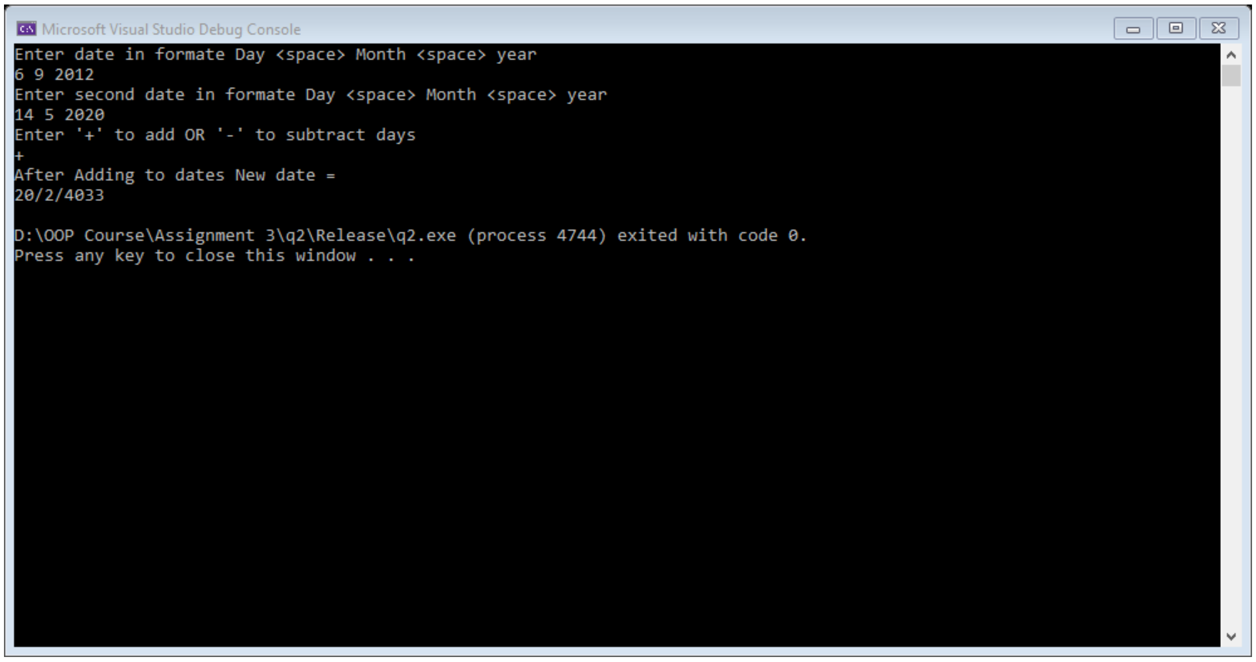
Cpp File

```
#include <iostream>
```

```cpp
#include "CalenderDate.h"
using namespace std;
int main()
{
	char choice;
	CalenderDate c1, c2,c3;
	int d = 0, m = 0, y = 0;
	cout << "Enter date in formate Day <space> Month <space> year" << endl;
	cin >> d >> m >> y;
	c1=CalenderDate(d, m, y);
	cout << "Enter second date in formate Day <space> Month <space> year " << endl;
	cin >> d >> m >> y;
	c2 = CalenderDate(d, m, y);
	cout << "Enter '+' to add OR '-' to subtract days" << endl;
	cin >> choice;
	switch (choice)
	{
	case '+':
		c3 = c2 += c1;
		c3.print();
		break;
	case '-':
		c3 = c2 -= c1;
		c3.print();
		break;
	default:
		break;
	}
}



CalenderDate::CalenderDate()
{
}


CalenderDate::~CalenderDate()
{
}

CalenderDate::CalenderDate(int da, int mon, int ye)
{
	days = da;
	months = mon;
	year = ye;
}
CalenderDate CalenderDate::operator+=(CalenderDate a)
{
	CalenderDate c;
	c.days = days + a.days;
	c.months = months  + a.months ;
	c.year = year + a.year;
	if (c.days > 30)
	{
		c.months = c.months + 1;
		c.days = c.days - 30;
	}
```

```
        if (c.months > 12)
        {
                c.year = c.year + 1;
                c.months = c.months - 12;
        }
        cout << "After Adding to dates New date =" << endl;
        return c;
}

CalenderDate CalenderDate::operator-=(CalenderDate b)
{
        CalenderDate c;
        c.days = days + b.days;
        c.months = months + b.months;
        c.year = year + b.year;
        if (c.days < 1)
        {
                c.months = c.months - 1;
                c.days = c.days + 30;
        }
        if (c.months < 1)
        {
                c.year = c.year - 1;
                c.months = c.months + 12;
        }
        cout << "After Subtracting two dates New date =" << endl;
        return c;
}
void CalenderDate::print()
{
        cout << days << '/' << months << '/' << year << endl;
```

Screen Shot



```
Enter date in formate Day <space> Month <space> year
6 9 2012
Enter second date in formate Day <space> Month <space> year
14 5 2020
Enter '+' to add OR '-' to subtract days
+
After Adding to dates New date =
20/2/4033

D:\OOP Course\Assignment 3\q2\Release\q2.exe (process 4744) exited with code 0.
Press any key to close this window . . .
```