CO3. Demonstrate a C++ code that creates a class called Fraction. The class Fraction has two attributes: numerator and denominator.

• In your constructor (inyour__init__ method), verify(assert?) that the numerator and denominator passed in during initiation are both of type int. If you want to be thorough, also check to make sure that the denominator is not zero.

• Write a .reduce() method that will reduce a fraction to lowest terms.

• Override the Object class's __str__ and __repl__ methods so that your objects will print out nicely. Remember that__str__ is more for humans; __repl__ is more for programmers. Ideally ,the__repl__ method will produce a string that you can run through the eval() function to clone the original fraction object.

• Override the + operator. In your code, this means that you will implement the special method __add__. The signature of the __add__ function will be def __add__(self, other): , and you'll return a new Fraction with the resultof the addition. Run your new Fraction through the reduce() function before returning**.**

code:

```
#include<iostream>

using namespace std;

class Fraction

{

public:

   int numerator,denominator;

   Fraction(int denominator,int numerator)

   {

      this->denominator=denominator;

      this->numerator=numerator;

   }

   int result()

   {
        float result=(float)denominator/numerator;

        cout<<"Fraction is :\t"<<denominator<<"/"<<numerator<<endl;

        cout<<result<<endl;

    }

    int gcd(int a, int b) {

   if (b == 0)

      return a;

      return gcd(b, a % b);

   }

   int reduce(int num1, int num2){

      int denom;

      denom = gcd(num1,num2);

      num1/=denom;

      num2/=denom;

      cout<< "Num1 = " << num1<<endl;

      cout<< "Num2 = " << num2<<endl;

      cout<< "Lowest Fraction : "<<num1<<"/"<<num2<<endl;

      }

      string str(string x)

      {

        return x;

      }

};

class new1

{

    public:
```

```
    string str(string x);
      void repl();
};
class Display:public new1
{
    public:
    string X;
    void repl( string x)
    {
        X=x;
    }
     void eval()
     {
        cout<<X<<endl;
     }
}d;
class add {
public:
    int j, l;
public:
    add(int r = 0, int i = 0) {j = r;   l = i;}
    add operator + (add const &obj) {
         add res;
         res.j = j + obj.j;
         res.l = l + obj.l;
         return res;
    }
    void print()
    {
        Fraction p(j,l);
        p.reduce(j,l);
         cout <<"Adition of first number"<< j <<endl<<"Addition of second number"<< l << '\n';
    }
};
int main()
{
    t:
        printf("ENTER DENOMINATOR :\t");
        int x;
    cin>>x;
    printf("ENTER NUMERATOR :\t");
        int y;
        cin>>y;
    if(y>1)
    {
         Fraction f(x,y);
         f.result();
        f.reduce(x,y);
    }
    else
    {
        goto t;
    }
    cout<<"\nYOUR COMMENT:\t";
    string x1;
    cin>>(x1);
    d.repl(x1);
    d.eval();
    cout<<"For ADDITION\n";
    cout<<"Enter the number 1:\t";
    cin>>x;
    cout<<"Enter the number 2:\t";
    cin>>y;
     add c1(x, y), c2(x, y);
    add c3 = c1 + c2;
    c3.print();
}
```