Follow Coding Standards

Day 5 Assignments

1. Write a class called date whose purpose is to convert a month and day into the absolute day of the year, ranging from 1 to 365. Ignore leap year, but check the month and day for valid data. As part of the class, include a static table that contains the number of days in each of the 12 months. Then write a program to test the class. Class definition is as given below.

class date {

static int table[12];

int day;

int month;

public :

date(int,int);

int convert();

};

Given the main() function

void main(){

date d1(10,5); int absday=0;

absday = d1.convert();

if(absday = = 0)

cout<<"\nWrong date is entered ";

else

cout<<"\nAbsolute date = "<<absday<<endl;

}

1. Given the following class definition that emulates a fractional number:

class fraction

{

private:

long num; //numerator

long den; //denominator

public:

// All member and friend functions

};

■ A constructor that accepts zero one or two arguments. The numerator defaults to zero, while the denominator defaults to 1. If one argument is specified, assume that it represents the numerator. Use only the member initialization list to construct the instance. If the denominator is equal to zero, change it to 1.

■ A copy constructor. Use only the member initialization list to construct the instance.

■ A function called print() that displays the fraction in its reduced form. In order to do this, call upon the global function gcd() that will return the greatest common denominator of two input numbers.

long gcd(long x, long y)

{

return (x == 0) ? y:gcd(y % x, x);

}

■ A friend function called add() that returns the sum of two fractions.

■ A friend function called subtract() that returns the difference two fractions.

■ A friend function called multiply() that returns the product of two fractions.

■ A friend function called divide() that returns the quotient of the first fraction divided by the second fraction.

■ An inline function called inc() that adds 1 to the invoking instance. Be sure to use the add() function and to allow for function chaining

Use the following main() function:

int main()

{

fraction f1, f2(2, 0);

fraction f3(f2);

f1.print();

f2.print();

f3.print();

f3=add(f3, fraction(-5, 4));

f1=add(f2, f3);

f1.print();

f1=sub(f2, f3);

f1.print();

f1=sub(f2, f3);

f1=print();

f1=mult(f2, f3);

f1.print();

f1.inc() .inc().print();

f1=div(f2, f3);

f1.print();

return 0;

}