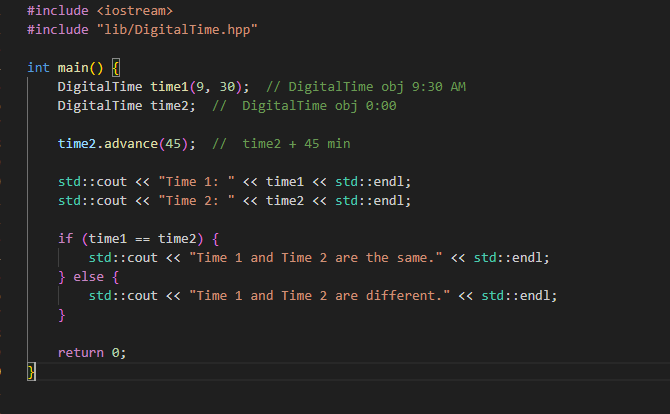
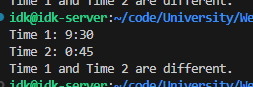
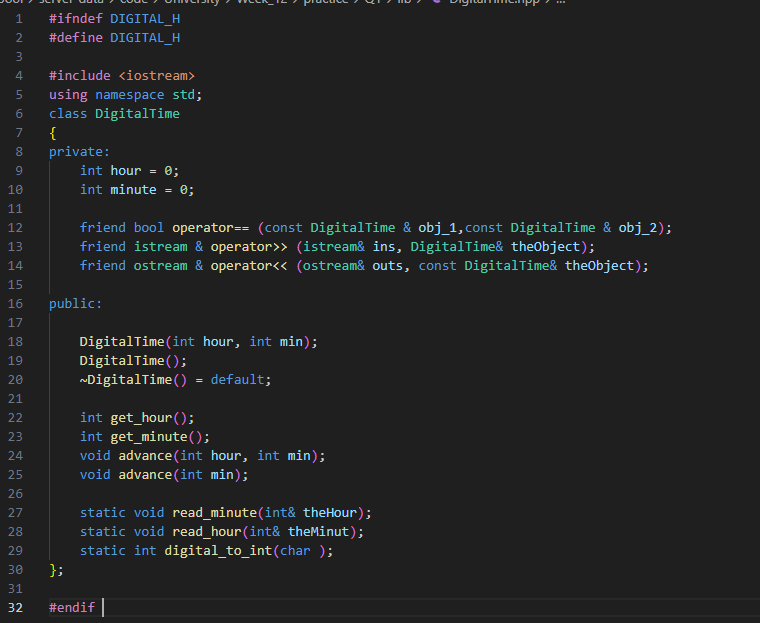
**Q1**

***./Q1.cpp***





***./lib/DigitalTime.hpp***



***./lib/DigitalTime.cpp***

//This is the implementation file: dtime.cpp of the class DigitalTime.

//The interface for the class DigitalTime is in the header file dtime.h.

// This is the same as 11-02.cpp

#include "DigitalTime.hpp"

#include <iostream>

#include <cctype>

#include <cstdlib>

using namespace std;

//Uses iostream and cstdlib:

DigitalTime::DigitalTime(int theHour, int theMinute)

{

    if (theHour < 0 || theHour > 24 || theMinute < 0 || theMinute > 59)

    {

        cout << "Illegal argument to DigitalTime constructor.";

        exit(1);

    }

    else

    {

        hour = theHour;

        minute = theMinute;

    }

    if (hour == 24)

        hour = 0; //standardize midnight as 0:00

}

DigitalTime::DigitalTime( )

{

    hour = 0;

    minute = 0;

}

int DigitalTime::get\_hour( )

{

    return hour;

}

int DigitalTime::get\_minute( )

{

    return minute;

}

void DigitalTime::advance(int minutesAdded)

{

    int grossMinutes = minute + minutesAdded;

    minute = grossMinutes%60;

    int hourAdjustment = grossMinutes/60;

    hour = (hour + hourAdjustment)%24;

}

void DigitalTime::advance(int hoursAdded, int minutesAdded)

{

    hour = (hour + hoursAdded)%24;

    advance(minutesAdded);

}

bool operator ==(const DigitalTime& time1, const DigitalTime& time2)

{

    return (time1.hour == time2.hour && time1.minute == time2.minute);

}

//Uses iostream:

ostream& operator <<(ostream& outs, const DigitalTime& theObject)

{

    outs << theObject.hour << ':';

    if (theObject.minute < 10)

        outs << '0';

    outs << theObject.minute;

    return outs;

}

//Uses iostream:

istream& operator >>(istream& ins, DigitalTime& theObject)

{

    DigitalTime::read\_hour(theObject.hour);

    DigitalTime::read\_minute(theObject.minute);

    return ins;

}

int DigitalTime::digital\_to\_int(char c)

{

    return ( int(c) - int('0') );

}

//Uses iostream, cctype, and cstdlib:

void DigitalTime::read\_minute(int& theMinute)

{

    char c1, c2;

    cin >> c1 >> c2;

    if (!(isdigit(c1) && isdigit(c2)))

    {

        cout << "Error illegal input to read\_minute\n";

        exit(1);

    }

    theMinute = digital\_to\_int(c1)\*10 + digital\_to\_int(c2);

    if (theMinute < 0 || theMinute > 59)

    {

        cout << "Error illegal input to read\_minute\n";

        exit(1);

    }

}

//Uses iostream, cctype, and cstdlib:

void DigitalTime::read\_hour(int& theHour)

{

    char c1, c2;

    cin >> c1 >> c2;

    if ( !( isdigit(c1) && (isdigit(c2) || c2 == ':' ) ) )

    {

        cout << "Error illegal input to read\_hour\n";

        exit(1);

    }

    if (isdigit(c1) && c2 == ':')

    {

        theHour = DigitalTime::digital\_to\_int(c1);

    }

    else //(isdigit(c1) && isdigit(c2))

    {

        theHour = DigitalTime::digital\_to\_int(c1)\*10

                  + DigitalTime::digital\_to\_int(c2);

        cin >> c2; //discard ':'

        if (c2 != ':')

        {

            cout << "Error illegal input to read\_hour\n";

            exit(1);

        }

    }

    if (theHour == 24)

        theHour = 0; //Standardize midnight as 0:00

    if ( theHour < 0 || theHour > 23 )

    {

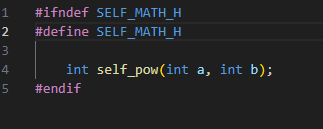
        cout << "Error illegal input to read\_hour\n";

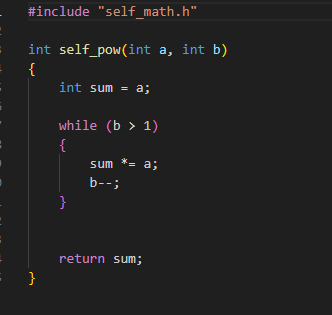
        exit(1);

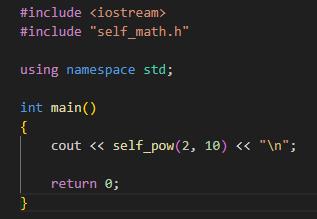
    }

}

**Q2**









**Q3**

