RollNo.22F-3298

Section-F

/\*Task-2 :

Create a structure called time. Its three members, all type int, should be called hours, minutes, and

seconds. Write a program that prompts the user to enter a time value in hours, minutes, and seconds.

This can be in 12:59:59 format, or each number can be entered at a separate prompt (“Enter hours:”,

and so forth). The program should then store the time in a variable of type struct time, and finally print

out the total number of seconds represented by this time value:

long totalsecs = t1.hours\*3600 + t1.minutes\*60 + t1.seconds\*/

#include <iostream>

using namespace std;

struct time

{

int hours;

int minutes;

int seconds;

};

int main()

{

struct time t1;

cout << "----------------Inputs-----------------" << endl;

cout << "Enter hours: ";

cin >> t1.hours;

cout << "Enter minutes: ";

cin >> t1.minutes;

cout << "Enter seconds: ";

cin >> t1.seconds;

cout << "---------------------Outputs-----------------------" << endl;

cout << "Total Time :" << endl;

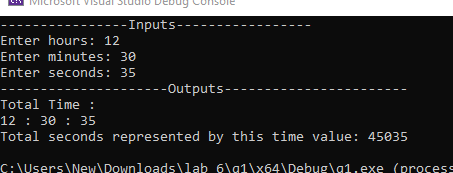
cout << t1.hours << " : " << t1.minutes << " : " << t1.seconds << endl;

long totalsecs = t1.hours \* 3600 + t1.minutes \* 60 + t1.seconds;

cout << "Total seconds represented by this time value: " << totalsecs << endl;

return 0;

}



/\*Task 3:

In the heyday of the British empire, Great Britain used a monetary system based on pounds, shillings, and

pence. There were 20 shillings to a pound, and 12 pence to a shilling. The notation for this old system used

the pound sign, £, and two decimal points, so that, for example, £5.2.8 meant 5 pounds, 2 shillings, and 8

pence. (Pence is the plural of penny.) The new monetary system, introduced in the 1950s, consists of only

pounds and pence, with 100 pence to a pound (like U.S. dollars and cents). We’ll call this new system decimal

pounds. Thus £5.2.8 in the old notation is £5.13 in decimal pounds (actually £5.1333333). Write a program to

convert the old pounds-shillings-pence format to decimal pounds. An example of the user’s interaction with

the program would be.

Enter pounds: 7

Enter shillings: 17

Enter pence: 9

Decimal pounds = £7.89

Create a structure called sterling that stores money amounts in the old-style British. The members could be

called pounds, shillings, and pence, all of type int. The program should ask the user to enter a money amount

in new-style decimal pounds (type double), convert it to the old-style system, store it in a variable of type

struct sterling, and then display this amount in pounds-shillings-pence format.\*/

#include <iostream>

using namespace std;

struct sterling

{

int pounds;

int shillings;

int pence;

};

int main()

{

double decimalPounds;

char c = 156; //Euro symbol in ASCII

cout << "-------------------Inputs--------------------------" << endl;

cout << "Enter Decimal pounds = "<<c;

cin >> decimalPounds;

sterling s;

s.pounds = decimalPounds;

decimalPounds = decimalPounds - s.pounds;

s.shillings = decimalPounds \* 20;

decimalPounds = decimalPounds \* 20;

decimalPounds = decimalPounds - s.shillings;

s.pence = decimalPounds \* 12;

cout << "-----------------------OutPuts--------------------------" << endl;

cout << "Pounds = " << s.pounds<<endl;

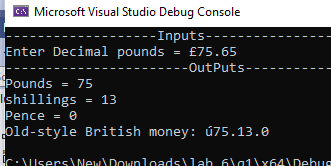
cout << "shillings = " << s.shillings<<endl;

cout << "Pence = " << s.pence<<endl;

cout << "Old-style British money: £" << s.pounds << "." << s.shillings << "." << s.pence << endl;

return 0;

}



/\*Task-04 :

Write a program to compare two dates entered by user. Make a structure named Date to store the elements

day, month and year to store the dates. If the dates are equal, display "Dates are equal" otherwise display

"Dates are not equal".\*/

#include <iostream>

using namespace std;

struct Date

{

int day;

int month;

int year;

};

int main()

{

Date date1, date2;

cout << "---------------------Input1-----------------------" << endl;

cout << "Enter first date: ";

cin >> date1.day;

cout << "Enter first month: ";

cin >> date1.month;

cout << "Enter first year: ";

cin>> date1.year;

cout << "---------------------Input2-----------------------" << endl;

cout << "Enter second date: ";

cin >> date2.day;

cout << "Enter second month: ";

cin >> date2.month;

cout << "Enter second year: ";

cin >> date2.year;

cout << "---------------------------------Outputs------------------" << endl;

// Comparison

if (date1.day == date2.day && date1.month == date2.month && date1.year == date2.year)

{

cout << "Dates are equal" << endl;

}

else

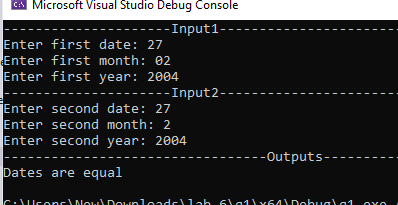
{

cout << "Dates are not equal" << endl;

}

return 0;

}



/\*Task-05 \*/

#include<iostream>

using namespace std;

class angle

{

private:

int degree;

float minute;

char direction;

public:

angle() {}

angle(int d,float m,char dir)

{

degree = d;

minute = m;

direction = dir;

}

int getdegree()

{

return degree;

}

void setAngle(int d, float m, char dir)

{

degree = d;

minute = m;

direction = dir;

}

void display()

{

cout << degree << '\xF8' << minute << "\' " << direction;

}

};

int main()

{

angle longitude(149, 34.8, 'N'), latitude(17, 21.5, 'S');

longitude.display();

cout << endl;

latitude.display();

return 0;

}

