Kelly Williams

11/12/23

ECE 1310.04

Homework 10

All code is in one solution at the end of the document.

1. 5.22 – *(Square of Asterisks)* Write a function that displays at the left margin of the screen a solid square of asterisks whole side is specified in integer parameter side.

A screenshot of a computer program

Description automatically generated

1. 5.25 – *(Calculating Number of Seconds)* Write a function that takes the time as three integer arguments (hours, minutes, seconds) and returns the number of seconds since the last time the clock “struck 12.” Use this function to calculate the amount of time in seconds between two times, both of which are within one 12-hour cycle of the clock.

A screenshot of a computer program

Description automatically generated

1. 5.32 – *(Quality Points for Numeric Grades)* Write a function qualityPoints that inputs a student’s average and returns a 4 if a student’s average is 90-100, 3 if the average is 80-89, 2 if the average is 70-79, 1 if the average is 60-69, and 0 if the average is lower than 60.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Code here:**

//System Libraries

#include <iostream>

//Function Prototypes

int validate(int, int, int);

void displaySquare(int);

int since12(int, int, int);

int qualityPoints(double);

//Global Constants

//Main Function

using namespace std;

int main(int argc, char\*\* argv)

{

int choice;

cout << "What would you like to run?\n"

<< "\t1. 5.22 - Square of Asterisks\n\t2. 5.25 - Calculating Number of Seconds\n\t3. 5.32 - Quality Points for Numeric Grades\n\n";

cin >> choice;

switch (choice)

{

case 1: //square of asterisks

{

cout << "\n\nSquare of Asterisks\n\n";

int side;

cout << "How long should the side of the square be?\n";

cin >> side;

cout << endl << endl << "Outputting square...\n\n";

displaySquare(side);

break;

}

case 2: //calculating number of seconds

{

cout << "\n\nCalculating Number of Seconds\n\n";

int hrs, mins, secs, between;

cout << "Input two times within the same 12-hour clock cycle.\n\n";

cout << "For the first time:\n"

<< "\nHours: ";

cin >> hrs;

hrs = validate(1, hrs, 12);

cout << "\nMinutes: ";

cin >> mins;

mins = validate(0, mins, 59);

cout << "\nSeconds: ";

cin >> secs;

secs = validate(0, secs, 59);

//get time1 in seconds

int time1 = since12(hrs, mins, secs);

cout << "\n\nFor the second time:\n"

<< "\nHours: ";

cin >> hrs;

hrs = validate(1, hrs, 12);

cout << "\nMinutes: ";

cin >> mins;

mins = validate(0, mins, 59);

cout << "\nSeconds: ";

cin >> secs;

secs = validate(0, secs, 59);

//get time2 in seconds

int time2 = since12(hrs, mins, secs);

between = time2 - time1;

cout << "\n\nSeconds elapsed between times: " << between << " seconds.\n";

break;

}

case 3: //quality points for numeric grades

{

cout << "\n\nQuality Points for Numeric Grades\n\n";

double avg;

int pts;

cout << "Student's average: ";

cin >> avg;

avg = validate(0, avg, 100);

pts = qualityPoints(avg);

cout << "This is worth " << pts << " quality points.\n";

break;

}

default:

{

cout << "\n\nwhat are you doing here?\n\n";

break;

}

}

return 0;

}

//input validation

int validate(int min, int val, int max)

{

while (val<min || val>max)

{

cout << "\nInvalid input! Must be greater than " << min << " but less than " << max << ". Try again.\n\n";

cin >> val;

}

return val;

}

//square of asterisks

void displaySquare(int side)

{

for (int i = 0; i < side; i++)

{

for (int j = 0; j < side; j++)

{

cout << "\*";

}

cout << endl;

}

}

//calculating number of seconds

int since12(int hrs, int mins, int secs)

{

int time = (hrs \* 3600) + (mins \* 60) + secs; //time in seconds

return time;

}

//quality points for numeric grades

int qualityPoints(double avg)

{

int points;

if (avg >= 90)

points = 4;

else if (90 > avg && avg >= 80)

points = 3;

else if (80 > avg && avg >= 70)

points = 2;

else if (70 > avg && avg >= 60)

points = 1;

else

points = 0;

cout << "points: " << points << endl;

return points;

}