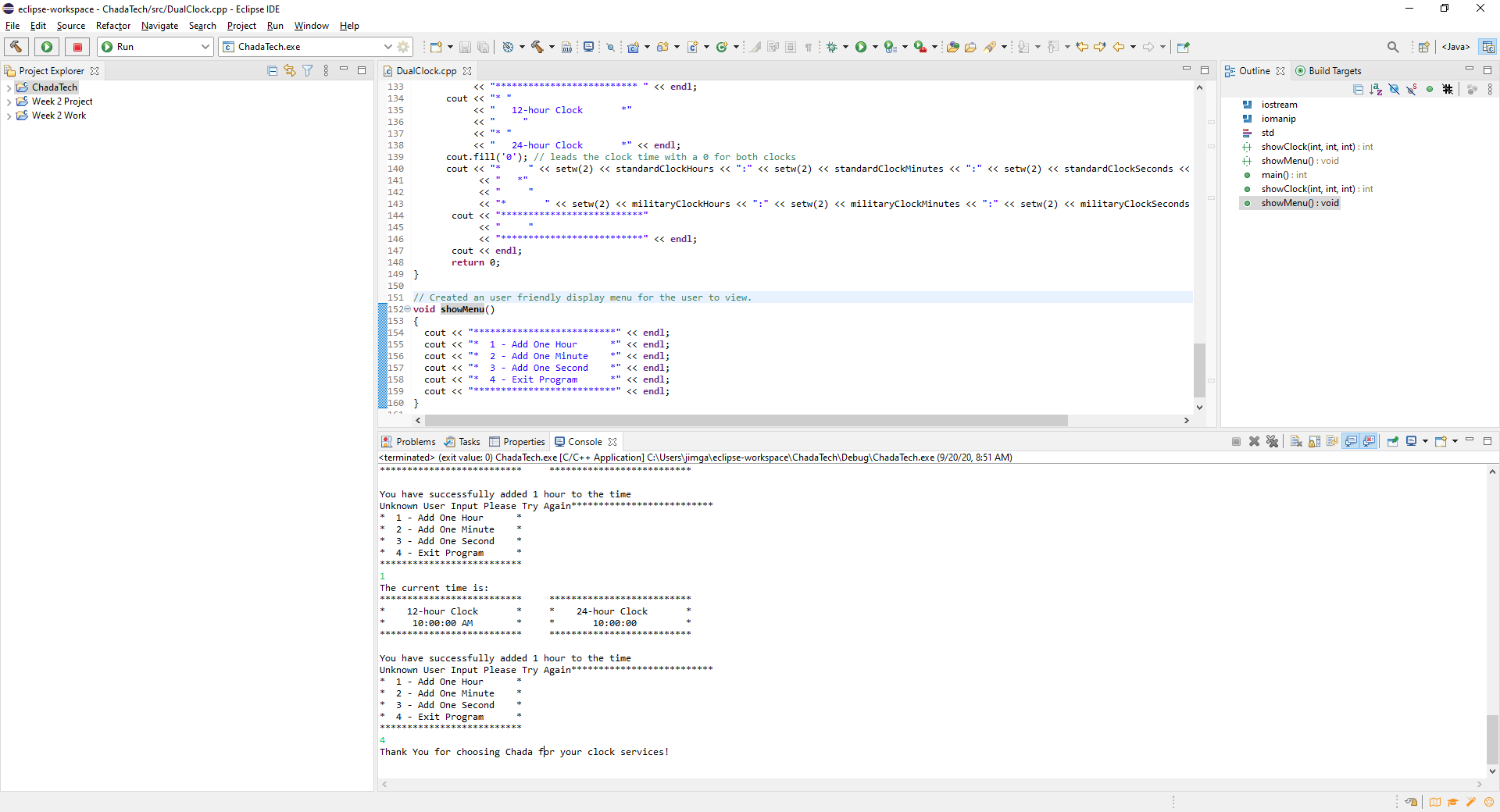
James Gabaree

SNHU



/\*

\* DualClock.cpp

\*

\* Created on: Sep 20, 2020

\* Author: James Gabaree

\* Email: James.Gabaree@snhu.edu

\*/

**# include** <iostream>

**# include** <iomanip>

**using** **namespace** std;

**int** **showClock**(**int**, **int**, **int**); // defined show clock in the format for hours, minutes, seconds.

**void** **showMenu**();

**int** **main** () // int chosen because main will return some type of output.

{

// creating variables to increase hours, minutes, and seconds by 1.

**int** increaseHour = 1;

**int** increaseMinute = 1;

**int** increaseSecond = 1;

//added default time in the format of int,int,int. Default time set to 0,0,0

showClock(0, 0, 0);

//displays options user can utilize

showMenu();

// initializing a userInput variable

**int** userInput = 0;

//takes the users input

cin >> userInput;

// created a loop so the user can select an option from the menu being 1-4

**while** (userInput !=4) {

**if** (userInput == 1) {

// allows the user to increase the hour by 1

showClock(increaseHour, 0, 0);

cout << "You have successfully added 1 hour to the time" << **endl**;

}

**if** (userInput == 2) {

// allows the user to increase the minutes by 1

showClock(0, increaseMinute, 0);

cout << "You have successfully added 1 minute to the time" << **endl**;

}

**if** (userInput == 3) {

showClock(0, 0, increaseSecond);

cout << "You have successfully added 1 second to the time" << **endl**;

}

**else** {

cout << "Unknown User Input Please Try Again";

}

showMenu(); // redisplays the correct options to the user.

cin >> userInput; // requests the user to correct the input

}

cout << "Thank You for choosing Chada for your clock services!" << **endl**; // thanks the user for utilizing our brand

**return** 0;

}

**int** **showClock**(**int** addHour, **int** addMinute, **int** addSecond) {

// adding variables to store clock time data for the 12 hour standard clock so time can be stored

**static** **int** standardClockHours = 0;

**static** **int** standardClockMinutes = 0;

**static** **int** standardClockSeconds = 0;

// standard 12 hour formatted time will need to display AM and PM. AM for before noon and PM for afternoon.

string beforeNoon = "AM";

string afterNoon = "PM";

//adding variables for 24 hours clock or military time clock so time can be stored

**static** **int** militaryClockHours = 0;

**static** **int** militaryClockMinutes = 0;

**static** **int** militaryClockSeconds = 0;

// creating a static string to store AM or PM. By default 0.0.0 is AM

**static** string amOrPm = beforeNoon;

//updated standardClock display and increasing hours

standardClockHours = standardClockHours + addHour;

standardClockMinutes = standardClockMinutes + addMinute;

standardClockSeconds = standardClockSeconds + addSecond;

//update military clock 24 display

militaryClockHours = militaryClockHours + addHour;

militaryClockMinutes = militaryClockMinutes + addMinute;

militaryClockSeconds = militaryClockSeconds + addSecond;

//check to see if hours in standard clock is past 12

**if** (standardClockHours > 12) {

**if** (amOrPm == beforeNoon)

amOrPm = afterNoon;

**else** // if Standard clock hours is less than 12 it will change to AM beforeNoon

amOrPm = beforeNoon;

// set the time back to 1 after it reaches 12 on the standard clock so it can be displayed correctly

standardClockHours = 1;

}

// creating clock reset logic for standard clock minutes to increase hours while returning to zero

**if** (standardClockMinutes > 59) {

// resets minutes back to zero and adds 1 hour to standard clock hours

standardClockMinutes = 0;

standardClockHours++;

}

// follow the same guidelines for seconds on the standard clock. reset seconds to 0 and increase minutes by 1

**if** (standardClockSeconds > 59) {

standardClockSeconds = 0;

standardClockMinutes++;

}

//updating military 24 hour clock

**if** (militaryClockHours > 24){

// clock is returned to 1

militaryClockHours = 1;

}

// will follow suite with defaulting minutes and seconds to 0 once they peak and adding the correct time to the corresponding time increment

**if** (militaryClockMinutes > 59){

militaryClockMinutes = 0;

militaryClockHours++;

}

**if** (militaryClockSeconds > 59){

militaryClockSeconds = 0;

militaryClockMinutes++;

}

// creating a working display for the user

cout << "The current time is: " << **endl**;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

<< " "

<< "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* " << **endl**;

cout << "\* "

<< " 12-hour Clock \*"

<< " "

<< "\* "

<< " 24-hour Clock \*" << **endl**;

cout.fill('0'); // leads the clock time with a 0 for both clocks

cout << "\* " << setw(2) << standardClockHours << ":" << setw(2) << standardClockMinutes << ":" << setw(2) << standardClockSeconds << " " << amOrPm << " "

<< " \*"

<< " "

<< "\* " << setw(2) << militaryClockHours << ":" << setw(2) << militaryClockMinutes << ":" << setw(2) << militaryClockSeconds << " \*" << **endl**;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

<< " "

<< "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << **endl**;

cout << **endl**;

**return** 0;

}

// Created an user friendly display menu for the user to view.

**void** **showMenu**()

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << **endl**;

cout << "\* 1 - Add One Hour \*" << **endl**;

cout << "\* 2 - Add One Minute \*" << **endl**;

cout << "\* 3 - Add One Second \*" << **endl**;

cout << "\* 4 - Exit Program \*" << **endl**;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << **endl**;

}