```
// Name        : SNHU Project 1: Chada Tech Clock App
// Author      : Jonathan Mitchell

#include <iostream>
#include <iomanip> // Allows use of setfill(), setw()

using namespace std;

//Clock variables using 24 hour time
int hours = 22; // hours variable can be changed to any # between 0 and 23
int minutes = 58; // minutes variable can be changed to any # between 0 and 59
int seconds = 56; // seconds variable can be changed to any # between 0 and 59

// Function that verifies if variable initialization numbers fall within acceptable
clock ranges
void DisplayTime(bool hour24) {
      if (hours < 0 || hours > 23 || minutes < 0 || minutes > 59 || seconds < 0 ||
            seconds > 59) {
            cout << "Invalid time entered.";
            return;
      }

      // Determine if hour corresponds with am or pm
      bool afterNoon = hours > 12;

      // Calculates 12 hour time by subtracting 12 hours from given hours variable
      int displayHours = afterNoon && !hour24 ? hours - 12 : hours;

      // Displays 00 for 12:00 am for 12 hour clock mode
      if (hours == 0 && !hour24) {
            displayHours = 12;
      }
      string ampm = "";

      /* For 12 hour clock, sets am/pm depending on if afterNoon indicates as
      true / false */
      if (!hour24) {
            ampm = afterNoon ? " PM" : " AM";
      }

      // Displays clock and fills 0 before hours between 0-9
      cout << setfill('0');
      cout << setw(2) << displayHours << ":";
      cout << setw(2) << minutes << ":";
      cout << setw(2) << seconds << ampm;
      cout << setfill(' ');
}

void DisplayClocks() {
      cout << flush;
      cout << "************************** **************************\n";
      cout << "*      12-Hour Clock      * *      24-Hour Clock      *\n";
      cout << "*       ";
      DisplayTime(false); //DisplayTime(false)
      cout << "       * *        ";
      DisplayTime(true); //DisplayTime(true)
      cout << "         *\n";
      cout << "************************** **************************\n";
}
```

```
/* After clock has been incremented based on MenuInput: If seconds is incremented,
corrects each clock so that 60 seconds
is reset to 00 & minutes is incremented by 1. Similarly, if minutes is incremented,
corrects each clock so that 60 minutes
is reset to 00 & hours is incremented by 1. If hours is incremented, corrects each
clock so that 24 hours is reset to 00.*/
void TimeFix() {
      if (seconds >= 60) {
            minutes++;
            seconds = seconds - 60;
      }
      if (minutes >= 60) {
            hours++;
            minutes = minutes - 60;
      }
      if (hours >= 24) {
            hours = hours - 24;
      }
}
bool MenuInput(char option) {
      // Adds 1 hour to both clocks
      if (option == '1') {
            hours++;
      }
      // Adds 1 minute to both clocks
      else if (option == '2') {
            minutes++;
      }
      // Adds 1 second to both clocks
      else if (option == '3') {
            seconds++;
      }
      // End program
      else if (option == '4') { // Ends program
            cout << "Program successfully exited. Goodbye!";
            return false;
      }
      // Increments seconds by 1 if option outside of menu selection is input
      else {
            seconds++;
      }
      /* continues loop through menu until user inputs option 4 & increments
      seconds by 1 */
      seconds++;
      return true;
}

bool DisplayMenu() {
      cout << "*************************\n";
      cout << "* 1 - Add One Hour       *\n";
      cout << "* 2 - Add One Minute     *\n";
      cout << "* 3 - Add One Second     *\n";
      cout << "* 4 - Exit Program       *\n";
      cout << "*************************\n";
      cout << "Which option would you like to select?" << endl;
      char option;      cin >> option;
      return MenuInput(option);
}
```

```
int main() {
	bool clocksOn = true;
	/* Loop that calls each function - displays both clocks & menu/asks for new
	menu inputand adjusts time based on selected time - until user selects to
exit the program*/
		while (clocksOn) {
			DisplayClocks();
			clocksOn = DisplayMenu();
			TimeFix();
			cin.clear(); // Clears previous input to avoid any potential
input issues
		}
	return 0;
}
```