This project will familiarize the student with the C++ class data structure. You will read data from the sequential file "my_input_file.txt" and execute the commands that are in this file. Lab 13 described the format of this file and meaning of each command.

The Time.h includes the following. This file is provided to you.

```
class Time {
public:
  Time();          // constuctor, set all attributes to 0
  ~Time() {};      // deconstructor
  void setTime(int h, int m, int s);  // command 1: set hour, min, sec
  void print();                       // command 2: print the time
  void setHour(int h);                // command 3: set hour
  void setMinute(int m);              // command 4: set minute
  void setSecond(int s);              // command 5: set second
  void incrementSecond();             // command 6: increment second
  void incrementMinute();             // command 7: increment minute
  void incrementHour();               // command 8: increment hour
  bool compareTime(Time t1);          // command 9: compare current
                                      // object with t1

private:
  int hour;     // 0 - 23    (24-hour clock format)
  int minute;   // 0 - 59
  int second;   // 0 - 59
};  // class Time
```

Use the provided template Time.cpp to define all the functions for the class.

Input file "my_input_file.txt" is provided to you. You can add additional commands to make sure you have exercised all the commands and your program works.

This lab consists of the following tasks.

1) Inside visual studio in your program, open input file "my_input_file.txt" for read only.

2) Open output file "my_output_file.txt" to write data into it. If the file "my_output_file.txt" does not exist then it will be created. If the file exists then its content will be deleted. A sample code is given below that you can use to create the output file.

```
ofstream outFile("my_output_file.txt", ios::out);
```

First print your name in capital letters to the output file.

3) Create two objects Time1 and Time2.

4) Read a command from the input file and if the command is not -1 then read other values needed by that command. Similar to lab 13, write into the output file the meaning of the command. Call the following functions when you encounter each of the commands below:

Command 1 (set time command): If object is 1 then call Time1.setTime(hour, min, sec). If object is 2 then call Time2.setTime(hour, min, sec).

Command 2 (print object): If object is 1 then call Time1.print(outFile), if object is 2 then call Time2.print(outFile).

Command 3 (set hour): If object is 1 then call Time1.setHour(hour). If object is 2 then call Time2.setHour(hour).

Command 4 (set minute): if object is 1 then call Time1.setMinute(min). If object is 2 then call Time2.setMinute(min).

Command 5 (set second): If object is 1 then call Time1.setSecond(sec). If object is 2 then call Time2.setSecond(sec).

Command 6 (increment second): if object is 1 then call Time1.incrementSecond(). If object is 2 then call Time2.incrementSecond().

Command 7 (increment minute): if object is 1 then call Time1.incrementMinute(). If object is 2 then call Time2. incrementMinute().

Command 8 (increment hour): if object is 1 then call Time1.incrementHour(). If object is 2 then call Time2. incrementHour().

Command 9 (compare time):
If object1 is 1 and object2 is 1 then call Time1.compareTime(Time1). If they match then write to the output file that "Object 1 and Object 1 match\n" otherwise write to the output file that "Object 1 and Object 1 do not match\n";

If object1 is 1 and object2 is 2 then call Time1.compareTime(Time2). If they match then write to the output file that "Object 1 and Object 2 match\n" otherwise write to the output file that "Object 1 and Object 2 do not match\n";

If object1 is 2 and object2 is 1 then call Time2.compareTime(Time1). If they match then write to the output file that "Object 2 and Object 1 match\n" otherwise write to the output file that "Object 2 and Object 1 do not match\n";

If object1 is 2 and object2 is 2 then call Time2.compareTime(Time2). If they match then write to the output file that "Object 2 and Object 2 match\n" otherwise write to the output file that "Object 2 and Object 2 do not match\n";

5) Repeat step 4 until you have processed all the commands. Note -1 indicates end of commands.

6) Close input file and output file and exit the program.

7) You can look at "my_output_file.txt" outside of the visual studio to see if the meaning of the commands have been successfully written to the output file and your program produces the correct result.

8) Submit the hard copy of following in your report:

a) my_input_file.txt
b) my_output_file.txt
c) lab14.cpp
d) Time.h
e) Time.cpp