CS22510

Assignment 1

Runners & Riders Out And About

Craig Heptinstall
3/22/2013

Contents

Introduction 2
1. Event Creator Program (C++) 2
(a) Printout of code 2
(b) Printout of attempting to compile program 2
(c) Printout of terminal session generated 3
(d) Generated files from program 5
2. Checkpoint Manager Program (Java) 6
(a) Printout of code 6
(b) Printout of attempting to compile program 6
(c) Printout of evidence of use of program 6
(i) File locking demonstration 11
3. Event Manager Program (C) 12
(a) Printout of attempting to compile program 12
(b) Printout of evidence of use of program 12
(c) Generated results from program 21
(d) Printout of log file 21
4. Program Descriptions 23
Event Creator 23
Checkpoint Manager 23
Event Manager 24
Appendix 1 26
Appendix 2 33

Introduction

This document outlines the solution for assignment one of this module. I have followed the brief handed to me and attempted to complete all sections to the best of my ability. I will follow all twelve sections required and place them in the subsections accordingly. The code for the event creation program along with the checkpoint manger will be presented in an appendix at the end of this report. These pages will also be page numbered (by hand). Any outputs or examples of the programs being run will be done by copying and pasting the terminal text under the correct headings and by using a grey text.

1. Event Creator Program (C++)

(a) Printout of code

See Appendix 1 for code of event creation program.

(b) Printout of attempting to compile program

I have below shoed the results of attempting to compile my program within NetBeans. I have simply copied this straight from the terminal window at the bottom of the IDE upon pressing the build and run button. As you can see, there are no errors or warnings detected by the compiler at this time.

```
"/usr/bin/make" -f nbproject/Makefile-Debug.mk QMAKE= SUBPROJECTS= .build-conf
make[1]: Entering directory
`/cygdrive/c/Users/Craig/Documents/NetBeansProjects/Event_Creation_crh13'
"/usr/bin/make"  -f nbproject/Makefile-Debug.mk dist/Debug/Cygwin-
Windows/event_creation_crh13.exe
make[2]: Entering directory
`/cygdrive/c/Users/Craig/Documents/NetBeansProjects/Event_Creation_crh13'
mkdir -p build/Debug/Cygwin-Windows
rm -f build/Debug/Cygwin-Windows/main.o.d
g++    -c -g -MMD -MP -MF build/Debug/Cygwin-Windows/main.o.d -o build/Debug/Cygwin-
Windows/main.o main.cpp
mkdir -p dist/Debug/Cygwin-Windows
g++     -o dist/Debug/Cygwin-Windows/event_creation_crh13 build/Debug/Cygwin-
Windows/main.o
make[2]: Leaving directory
`/cygdrive/c/Users/Craig/Documents/NetBeansProjects/Event_Creation_crh13'
make[1]: Leaving directory
`/cygdrive/c/Users/Craig/Documents/NetBeansProjects/Event_Creation_crh13'


BUILD SUCCESSFUL (total time: 18s)
```

(c) Printout of terminal session generated

I have below shown my event creation program running within the NetBeans terminal. To show the program running properly, I have ensured that every functionality was tested, including the creation of an event, several entrants and several courses. I have also shown an example of trying to register a non-existent node to a course.

```
*******************************************
*                                         *
*          Please Select An Option:       *
*-----------------------------------------*
*  1.Enter An Event                       *
*  2.Enter The Entrants For Event         *
*  3.Enter Courses Nodes For Event        *
*-----------------------------------------*
*  4.Quit                                 *
*******************************************
1
Please Input Event Name: Bike Road Race
Please Input Event Date(e.g 25th June 2012): 20th May 2013
Please Input Event Time(e.g 09:10): 09:30


*******************************************
*                                         *
*          Please Select An Option:       *
*-----------------------------------------*
*  1.Enter An Event                       *
*  2.Enter The Entrants For Event         *
*  3.Enter Courses Nodes For Event        *
*-----------------------------------------*
*  4.Quit                                 *
*******************************************


2
How Many Entrants Are There?: 7
Please Enter Entrants Full Name: Craig Heptinstall
Please Enter Entrants Course: D
Please Enter Entrants Full Name: Jim Terry
Please Enter Entrants Course: C
Please Enter Entrants Full Name: Robert Junior JR
Please Enter Entrants Course: B
Please Enter Entrants Full Name: Test Subject
Please Enter Entrants Course: D
Please Enter Entrants Full Name: Tim East Dr
Please Enter Entrants Course: D
Please Enter Entrants Full Name: Craig Rogrs
Please Enter Entrants Course: C
Please Enter Entrants Full Name: Test Man
Please Enter Entrants Course: B
```

```
*********************************************
*                                           *
*          Please Select An Option:         *
*-------------------------------------------*
*  1.Enter An Event                         *
*  2.Enter The Entrants For Event           *
*  3.Enter Courses Nodes For Event          *
*-------------------------------------------*
*  4.Quit                                   *
*********************************************
```

```
3
Input Filename For Nodes: nodes.txt
How Many Courses Do You Want To Input?: 3
Enter Course Letter: B
How Many Nodes Are On This Course?: 7
Input Node: 1:1
Input Node: 2:2
Input Node: 3:3
Input Node: 4:6
Input Node: 5:4
Input Node: 6:3
Input Node: 7:1
Enter Course Letter: C
How Many Nodes Are On This Course?: 9
Input Node: 1:1
Input Node: 2:3
Input Node: 3:5
Input Node: 4:7
Input Node: 5:8
Input Node: 6:11
Input Node: 7:8
Input Node: 8:4
Input Node: 9:1
Enter Course Letter: D
How Many Nodes Are On This Course?: 5
Input Node: 1:1
Input Node: 2:5
Input Node: 3:9
Input Node: 4:14
Node Does Not Exist. Try Again.Input Node: 4:12
Input Node: 5:9
```

```
*********************************************
*                                           *
*          Please Select An Option:         *
*-------------------------------------------*
*  1.Enter An Event                         *
*  2.Enter The Entrants For Event           *
*  3.Enter Courses Nodes For Event          *
*-------------------------------------------*
*  4.Quit                                   *
*********************************************
```

4

RUN SUCCESSFUL (total time: 6m 32s)

(d) Generated files from program

Now I have displayed the run time of the event creation program, I will now display the text files that were created as a result of the functions. As you will see, the files have correctly outputted to the standard they were presented to me as example files, and all reflect the data inputted in the creation program. These are now ready to be read into and used in the checkpoint manager and event manager programs.

In “entrants.txt”:

1 D Craig Heptinstall
2 C Jim Terry
3 B Robert Junior JR
4 D Test Subject
5 D Tim East Dr
6 C Craig Rogrs
7 B Mr Test Man

In “name.txt”:

Bike Road Race
20th May 2013
09:30

In “courses.txt”:

B 7 1 2 3 6 4 3 1
C 9 1 3 5 7 8 11 8 4 1
D 5 1 5 9 12 9

2. Checkpoint Manager Program (Java)

(a) Printout of code

See Appendix 2 for code of checkpoint manager program

(b) Printout of attempting to compile program

Below displays the attempt to compile my program as the 'build and clean' option in NetBeans. As you will see, there appears to be one warning from the terminal. I will cover this in my report at the end of this document.

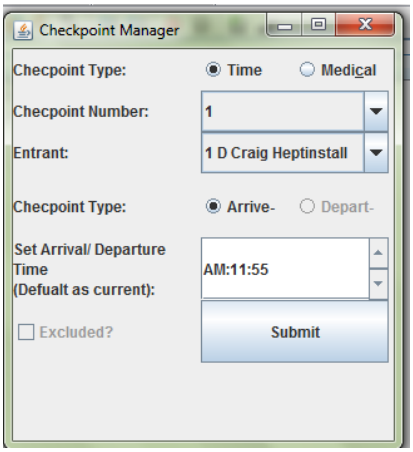
```
ant -f C:\\Users\\Craig\\Dropbox\\Checkpoint_Manager_crh13 clean jar
init:
deps-clean:
Updating property file:
C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build\built-clean.properties
Deleting directory C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build
clean:
init:
deps-jar:
Created dir: C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build
Updating property file:
C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build\built-jar.properties
Created dir: C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build\classes
Created dir: C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build\empty
Created dir: C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build\generated-
sources\ap-source-output
Compiling 5 source files to
C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build\classes
Note:
C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\src\uk\ac\aber\dcs\crh13\panel\
Panel.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Copied 1 empty directory to 1 empty directory under
C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build\classes
compile:
Created dir: C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\dist
Copying 1 file to C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\build
Nothing to copy.
Building jar:
C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\dist\Checkpoint_Manager_crh1
3.jar
To run this application from the command line without Ant, try:
java -jar
"C:\Users\Craig\Dropbox\Checkpoint_Manager_crh13\dist\Checkpoint_Manager_crh
13.jar"
jar:
BUILD SUCCESSFUL (total time: 12 seconds)
```

(c) Printout of evidence of use of program

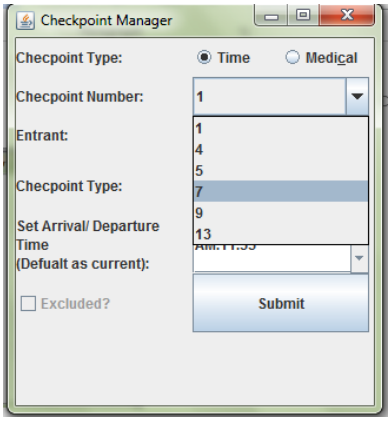
In this section of the document I will now display my checkpoint manager program at run time, displaying the input of the required files at command line at the start of the program, with the addition of an incorrect entry to demonstrate the file exists. I will then move onto the graphical user interface side of my program and demonstrate the different options available for registering check point times for entrants. For this demonstration, I will be using a txt file already containing some

information about my entrants check point information to show how these can be apprehended to by the program. I will also be using the same course and entrants file produced in the previous program.

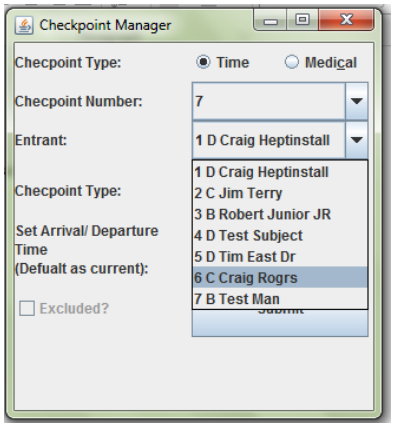
```
  ____ _                 _                 _       _
 / ___\ |__   ____  ____| | ____ __  ____ (_)_ __ | |_
/ /   | '_ \ / _ \/ __| |/ / '_ \ / _ \| | '_ \| __|
/ /____| | | |  __/ (__|   <| |_) | (_) | | | | | |_
\_____/|_| |_|\____|\___|_|\_\ .__/ \____/|_|_| |_|\__|
                             |_|
 /\/\   __ _ _ __   __ _  __ _  ___ _ __
/    \ / _` | '_ \ / _` |/ _` |/ _ \ '__|
/ /\/\ \ (_| | | | | (_| | (_| |  __/ |
\/    \/\__,_|_| |_|\__,_|\__, |\___|_|
                           |___/
(crh13)

Please Input Entrants File Name:
entrants.txt

Please Input Nodes File Name:
nodes,dsfs
No or Incorrect File Inputted


Please Input Nodes File Name:
nodes.txt

Please Input Courses File Name:
courses.txt

Please Input The Checkpoint Times File Name You Want To Update:
cp_times_1.txt
```



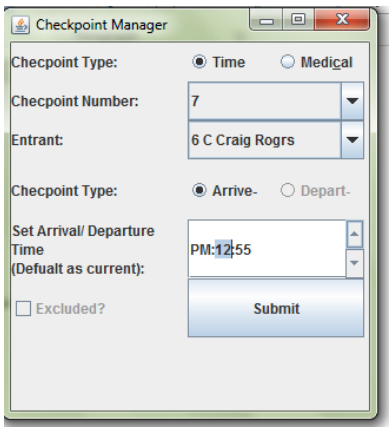
This shot shows the graphical user interface when first loaded up. As you can see, it has the first option to choose what type of checkpoint it is. When 'time' is selected, then the depart and excluded options are not available. This removes any need for extra checking.



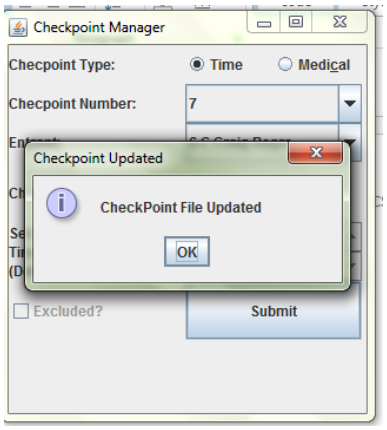
In this shot I have illustrated the checkpoint selection box, which I have made so it only contains registered checkpoints from the nodes file. Again, this removes any further need for any extra checking when submitting a checkpoint number.



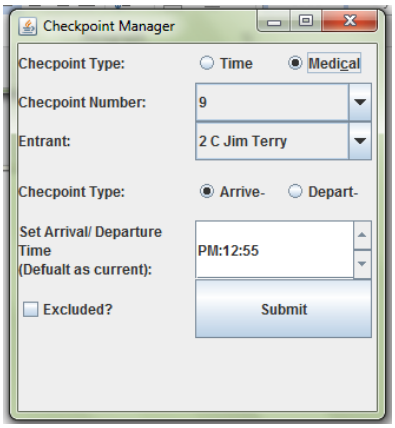
Shown here is the list of entrants gained from the input of the entrants file earlier. Placed into a combo box, I have included the name, number and course letter to make it easier for users to recognise a specific entrant.



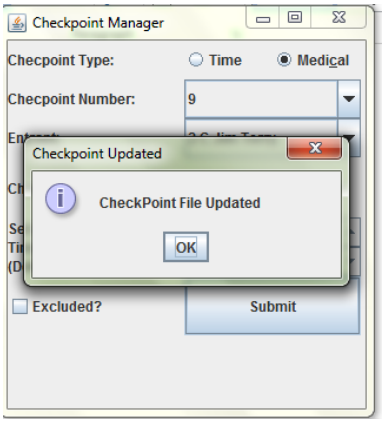
Concerning the input of the time, I chose to use a JSpinner box, which allows the user to simply click on the am/pm, hour or minutes and then click up or down on the arrows to change the time. It also avoids the need for validation because if the user types in an incorrect time it would simply revert to the previous time. This Box always begins at the default time which is the current time.



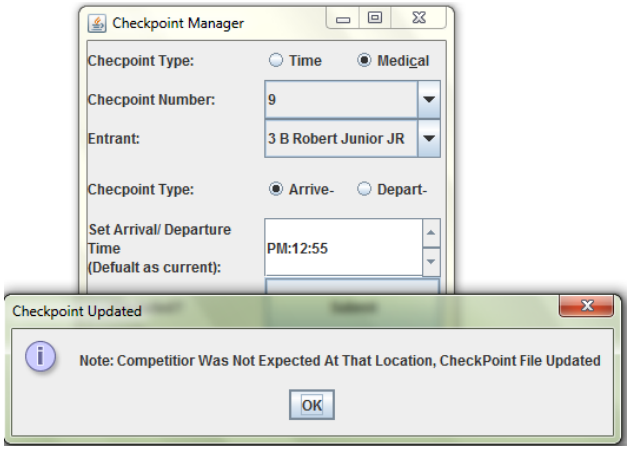
Once the user has entered the required data for a checkpoint, they then simply click the submit button which confirms the checkpoint being appended to the checkpoint file.



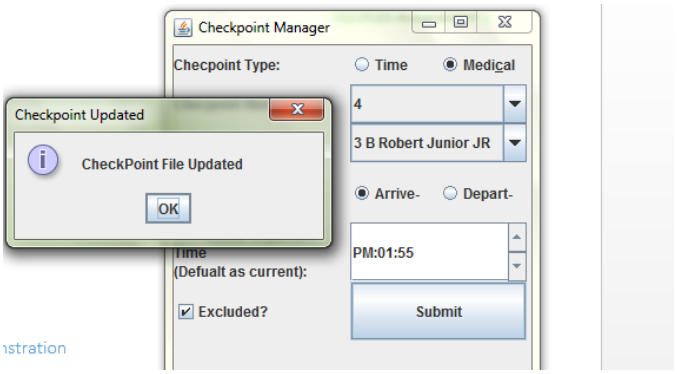
In the next example of a different checkpoint time, I have selected a medical type and changed the entrant number and checkpoint number. As you now see, the depart and excluded options are available from the GUI.



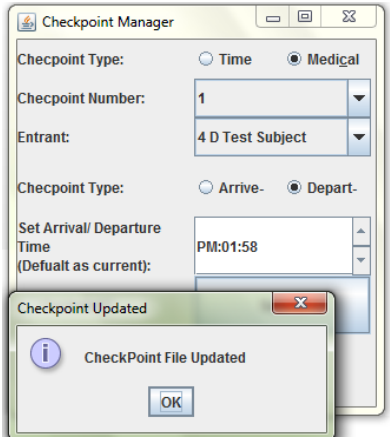
Clicking the submit button again showed a notification that the checkpoint file had been updated.



Here I have shown an example of the check performed to see if the checkpoint number an entrant is at is one on their course. In this example, the checkpoint number for this entrant does not appear to be on the entrant's course and has written this to the checkpoint file and notified the user.



In another example, I have shown the use of the 'excluded' button to exclude an entrant. I have also changed the time to reflect this happening at a later time of the day.



Finally, to ensure I have shown all of the checkpoint options, I have selected medical again, but with the 'depart' option selected to signify the entrant left a specific checkpoint at the time chosen.

After times were inputted using checkpoint manager program, I can now show what output was placed in the checkpoint file chosen. This reflects the screenshots presented earlier. Where I showed a simple time checkpoint it has been indicated with a 'T, where it was a medical arrival checkpoint it has been shown with an 'A', for an incorrect checkpoint it has been 'I', excluded as indicated by the 'E' and for a departed checkpoint time it has been indicated by a 'D'.

T 7 6 12:55
A 9 2 12:55
I 9 3 12:55
E 4 3 13:55
D 4 3 13:58

(i) File locking demonstration

In this short section of the document branching off from the demonstration of my checkpoint manager program, I am going to show simply how file locking works when running my program form the user's point of view.

```
  ____ _                 _                 _       _
 / ___\ |__   ____  ____| | ____ __   ____ (_)_ __ | |_
/ /   | '_ \ / _ \/ ___| |/ / '_ \ / _ \| | '_ \| __|
/ /___| | | |  __/ (___|   <| |_) | (_) | | | | | |_
\_____/|_| |_|\____|\____|_|\_\ .__/ \____/|_|_| |_|\__|
                              |_|
 /\/\   __ _ _ __   __ _  __ _  ___ _ __
/    \ / _` | '_ \ / _` |/ _` |/ _ \ '__|
/ /\/\ \ (_| | | | | (_| | (_| |  __/ |
\/    \/\__,_|_| |_|\__,_|\__, |\___|_|
                           |___/
(crh13)

Please Input Entrants File Name:
entrants.txt

Please Input Nodes File Name:
nodes.txt

Please Input Courses File Name:
courses.txt

Please Input The Checkpoint Times File Name You Want To Update:
cp_times_1.txt

File Is Currently Locked Try Again Later

Please Input The Checkpoint Times File Name You Want To Update:
```

The easiest way to demonstrate my programs file looking feature was simply to run the program in two different IDE's and then attempt to run the program in one before the other had the chance to try and lock the checkpoints file. As you can see in the above print of the terminal, this was for the NetBeans version run after the eclipse version. As shown, the system tells the user that the file is currently locked and asks the user to try again. The user would simply wait here and try get into the checkpoint file when it is free.

3. Event Manager Program (C)

(a) Printout of attempting to compile program

```
"/usr/bin/make" -f nbproject/Makefile-Debug.mk QMAKE= SUBPROJECTS= .build-
conf
make[1]: Entering directory
`/cygdrive/c/Users/Craig/Documents/NetBeansProjects/Event_Manager_crh13'
"/usr/bin/make"  -f nbproject/Makefile-Debug.mk dist/Debug/Cygwin-
Windows/event_manager_crh13.exe
make[2]: Entering directory
`/cygdrive/c/Users/Craig/Documents/NetBeansProjects/Event_Manager_crh13'
mkdir -p build/Debug/Cygwin-Windows
rm -f build/Debug/Cygwin-Windows/main.o.d
gcc    -c -g -MMD -MP -MF build/Debug/Cygwin-Windows/main.o.d -o
build/Debug/Cygwin-Windows/main.o main.c
main.c: In function `logFunction':
main.c:221: warning: passing arg 2 of `fprintf' makes pointer from integer without a
cast
mkdir -p dist/Debug/Cygwin-Windows
gcc     -o dist/Debug/Cygwin-Windows/event_manager_crh13 build/Debug/Cygwin-
Windows/supply.o  build/Debug/Cygwin-Windows/entrants.o  build/Debug/Cygwin-
Windows/excluded.o  build/Debug/Cygwin-Windows/main.o  build/Debug/Cygwin-
Windows/data.o
make[2]: Leaving directory
`/cygdrive/c/Users/Craig/Documents/NetBeansProjects/Event_Manager_crh13'
make[1]: Leaving directory
`/cygdrive/c/Users/Craig/Documents/NetBeansProjects/Event_Manager_crh13'


BUILD SUCCESSFUL (total time: 6s)
```

(b) Printout of evidence of use of program

In the following section I will be displaying the runtime of my event manager program, using the files created in the event creation program, and using the checkpoint file created in the checkpoint manager program. Below, I have shown the use of every query and function once, and shown the output for each. After this section, I will go back to my checkpoint manager program and make it so that a number of entrants appear to be finished. This will allow me to then use the results table option in the event manger program to display a complete results table. Any error checking of the users inputs integrated within my program are also displayed in the example run below.

```
  __  ____  _____   _  __   _____ _______  _____    _____
 /  |/  / / __  )  / |/ /  / _/ /_  __/ / __ (   / _ (
/ /|_/ / / / / /  /  |/ /   / /    / /   / / / / / /_/ /
/ /  / / / /_/ /  / /| /   _/ /    / /   / /_/ / / _, _/
/_/  /_/  (_____/  /_/ |_/  /___/   /_/    (____/ /_/ |_|
                                                (Extended Version)

Enter file name for event name: name.txt
Bike Road Race
20th May 2013
09:30
Enter file name for nodes: nodes.txt
1 CP
2 JN
```

```
3 JN
4 CP
5 CP
6 JN
7 CP
8 JN
9 CP
10 JN
11 JN
12 JN
13 CP
14 CP
Enter file name for tracks: tracks.txt
1 1 2 20
2 2 3 10
3 3 4 11
4 4 5 15
5 5 6 12
6 6 8 10
7 6 7 8
8 7 10 12
9 8 10 10
10 8 9 5
11 3 9 18
12 9 12 20
13 2 13 30
14 12 13 5
15 10 11 15
16 11 12 5
16 11 12 5
Enter file name for courses: courses.txt
B 7 1 2 3 6 4 3 1
C 9 1 3 5 7 8 11 8 4 1
D 5 1 5 9 12 9
Enter file name for entrants: entrants.txt
1 D Craig Heptinstall
2 C Jim Terry
3 B Robert Junior JR
4 D Test Subject
5 D Tim East Dr
6 C Craig Rogrs
7 B Test Man
```

```
Welcome to system developed for:
   Bike Road Race
   20th May 2013
   09:30

    ******************************************************
    *                                                    *
    *           Please Select An Option:                 *
    *----------------------------------------------------*
    *  1.Locate Competitor                               *
    *  2.Count Competitors That Have Not Started         *
    *  3.Count Competitors On Course                     *
    *  4.Count Competitors Finished                      *
    *  5.Manually Supply Times                           *
    *  6.Supply Checkpoint Files                         *
    *  7.Print Results                                   *
    *  8.Supply Medical Checkpoint Times                 *
    *  9.List Those Excluded For Wrong Routes            *
    * 10.List Those Excluded For Medical Reasons         *
    *----------------------------------------------------*
    * 11.Quit                                           *
    ******************************************************
6
Enter file name for checkpoint file: cp_times_1.txt

T 7 6 12:55
A 9 2 12:55
I 9 3 12:55
E 4 3 13:55
D 4 3 13:58
Type 'c' to continue...c

    ******************************************************
    *                                                    *
    *           Please Select An Option:                 *
    *----------------------------------------------------*
    *  1.Locate Competitor                               *
    *  2.Count Competitors That Have Not Started         *
    *  3.Count Competitors On Course                     *
    *  4.Count Competitors Finished                      *
    *  5.Manually Supply Times                           *
    *  6.Supply Checkpoint Files                         *
    *  7.Print Results                                   *
    *  8.Supply Medical Checkpoint Times                 *
    *  9.List Those Excluded For Wrong Routes            *
    * 10.List Those Excluded For Medical Reasons         *
    *----------------------------------------------------*
```

```
*  11.Quit                                          *
****************************************************
1

Please Input Competitor Number: 2

Competitor 2, Jim Terry has arrived at a medical checkpoint on Course C, Current
time 12:55

Type 'c' to continue...c




****************************************************
*                                                  *
*           Please Select An Option:               *
*--------------------------------------------------*
*  1.Locate Competitor                             *
*  2.Count Competitors That Have Not Started       *
*  3.Count Competitors On Course                   *
*  4.Count Competitors Finished                    *
*  5.Manually Supply Times                         *
*  6.Supply Checkpoint Files                       *
*  7.Print Results                                 *
*  8.Supply Medical Checkpoint Times               *
*  9.List Those Excluded For Wrong Routes          *
* 10.List Those Excluded For Medical Reasons       *
*--------------------------------------------------*
* 11.Quit                                          *
****************************************************
1

Please Input Competitor Number: 1

Competitor 1, Craig Heptinstall has not started yet- Course D

Type 'c' to continue...c




****************************************************
*                                                  *
*           Please Select An Option:               *
*--------------------------------------------------*
*  1.Locate Competitor                             *
```

```
*  2.Count Competitors That Have Not Started       *
*  3.Count Competitors On Course                  *
*  4.Count Competitors Finished                  *
*  5.Manually Supply Times                       *
*  6.Supply Checkpoint Files                     *
*  7.Print Results                              *
*  8.Supply Medical Checkpoint Times            *
*  9.List Those Excluded For Wrong Routes       *
* 10.List Those Excluded For Medical Reasons     *
*-----------------------------------------------*
* 11.Quit                                       *
*************************************************
```

2

Number of Competitors not Started: 4

Type 'c' to continue...c

```
*************************************************
*                                               *
*           Please Select An Option:            *
*-----------------------------------------------*
*  1.Locate Competitor                          *
*  2.Count Competitors That Have Not Started       *
*  3.Count Competitors On Course                  *
*  4.Count Competitors Finished                  *
*  5.Manually Supply Times                       *
*  6.Supply Checkpoint Files                     *
*  7.Print Results                              *
*  8.Supply Medical Checkpoint Times            *
*  9.List Those Excluded For Wrong Routes       *
* 10.List Those Excluded For Medical Reasons     *
*-----------------------------------------------*
* 11.Quit                                       *
*************************************************
```

3

Number of Competitors on Course: 2

Type 'c' to continue...c

```
*******************************************************
*                                                     *
*           Please Select An Option:                  *
*-----------------------------------------------------*
*  1.Locate Competitor                                *
*  2.Count Competitors That Have Not Started          *
*  3.Count Competitors On Course                      *
*  4.Count Competitors Finished                       *
*  5.Manually Supply Times                            *
*  6.Supply Checkpoint Files                          *
*  7.Print Results                                    *
*  8.Supply Medical Checkpoint Times                  *
*  9.List Those Excluded For Wrong Routes             *
* 10.List Those Excluded For Medical Reasons          *
*-----------------------------------------------------*
* 11.Quit                                             *
*******************************************************
4

Number of Competitors Finished: 0

Type 'c' to continue...c




*******************************************************
*                                                     *
*           Please Select An Option:                  *
*-----------------------------------------------------*
*  1.Locate Competitor                                *
*  2.Count Competitors That Have Not Started          *
*  3.Count Competitors On Course                      *
*  4.Count Competitors Finished                       *
*  5.Manually Supply Times                            *
*  6.Supply Checkpoint Files                          *
*  7.Print Results                                    *
*  8.Supply Medical Checkpoint Times                  *
*  9.List Those Excluded For Wrong Routes             *
* 10.List Those Excluded For Medical Reasons          *
*-----------------------------------------------------*
* 11.Quit                                             *
*******************************************************
5

Please Input Competitor Number: 6

Enter a Time Checkpoint number the Competitor has Reached: 9
CP
Enter Current Time for Competitor (e.g  11:20): 13:30
Type 'c' to continue...c
```

```
*******************************************************
*                                                     *
*           Please Select An Option:                  *
*-----------------------------------------------------*
*  1.Locate Competitor                                *
*  2.Count Competitors That Have Not Started          *
*  3.Count Competitors On Course                      *
*  4.Count Competitors Finished                       *
*  5.Manually Supply Times                            *
*  6.Supply Checkpoint Files                          *
*  7.Print Results                                    *
*  8.Supply Medical Checkpoint Times                  *
*  9.List Those Excluded For Wrong Routes             *
* 10.List Those Excluded For Medical Reasons          *
*-----------------------------------------------------*
* 11.Quit                                             *
*******************************************************
8

Please Input Competitor Number: 2
Sorry, Competitor Excluded.
Type 'c' to continue...c
```

```
*******************************************************
*                                                     *
*           Please Select An Option:                  *
*-----------------------------------------------------*
*  1.Locate Competitor                                *
*  2.Count Competitors That Have Not Started          *
*  3.Count Competitors On Course                      *
*  4.Count Competitors Finished                       *
*  5.Manually Supply Times                            *
*  6.Supply Checkpoint Files                          *
*  7.Print Results                                    *
*  8.Supply Medical Checkpoint Times                  *
*  9.List Those Excluded For Wrong Routes             *
* 10.List Those Excluded For Medical Reasons          *
*-----------------------------------------------------*
* 11.Quit                                             *
*******************************************************
```

```
8

Please Input Competitor Number: 1

Enter the Medical Checkpoint number the Competitor has Reached/ Left: 1

Left or Arrived? ('l'- left    'a'- arrived )l

Enter Current Time for Competitor (e.g  11:20): 12:58
Type 'c' to continue...c
```

```
*******************************************************
*                                                     *
*           Please Select An Option:                  *
*-----------------------------------------------------*
*  1.Locate Competitor                                *
*  2.Count Competitors That Have Not Started          *
*  3.Count Competitors On Course                      *
*  4.Count Competitors Finished                       *
*  5.Manually Supply Times                            *
*  6.Supply Checkpoint Files                          *
*  7.Print Results                                    *
*  8.Supply Medical Checkpoint Times                  *
*  9.List Those Excluded For Wrong Routes             *
* 10.List Those Excluded For Medical Reasons          *
*-----------------------------------------------------*
* 11.Quit                                             *
*******************************************************
9
Excluded for wrong course:
   3 Robert Junior JR

Type 'c' to continue...c
```

```
*******************************************************
*                                                     *
*           Please Select An Option:                  *
*-----------------------------------------------------*
*  1.Locate Competitor                                *
*  2.Count Competitors That Have Not Started          *
*  3.Count Competitors On Course                      *
```

```
*  4.Count Competitors Finished                  *
*  5.Manually Supply Times                       *
*  6.Supply Checkpoint Files                     *
*  7.Print Results                               *
*  8.Supply Medical Checkpoint Times             *
*  9.List Those Excluded For Wrong Routes        *
* 10.List Those Excluded For Medical Reasons     *
*-----------------------------------------------*
* 11.Quit                                       *
*************************************************
10
Excluded for safety reasons:


Type 'c' to continue...c




*************************************************
*                                               *
*           Please Select An Option:            *
*-----------------------------------------------*
*  1.Locate Competitor                          *
*  2.Count Competitors That Have Not Started    *
*  3.Count Competitors On Course                *
*  4.Count Competitors Finished                 *
*  5.Manually Supply Times                      *
*  6.Supply Checkpoint Files                    *
*  7.Print Results                              *
*  8.Supply Medical Checkpoint Times            *
*  9.List Those Excluded For Wrong Routes       *
* 10.List Those Excluded For Medical Reasons    *
*-----------------------------------------------*
* 11.Quit                                       *
*************************************************

11

RUN SUCCESSFUL (total time: 9m 0s)
```

(c) Generated results from program

Now I have shown each function being used in the event manager program except the print results function, I will now go back to my checkpoint manager program and ensure that all competitors except for those who are excluded complete their course. Once this is complete, I will then display the generated results table below. I will discuss the table print function in my program description further to explain how the this function performed. The function being run from the menu is shown including the generated results table:

```
*****************************************************
*                                                   *
*           Please Select An Option:                *
*---------------------------------------------------*
*  1.Locate Competitor                              *
*  2.Count Competitors That Have Not Started        *
*  3.Count Competitors On Course                    *
*  4.Count Competitors Finished                     *
*  5.Manually Supply Times                          *
*  6.Supply Checkpoint Files                        *
*  7.Print Results                                  *
*  8.Supply Medical Checkpoint Times                *
*  9.List Those Excluded For Wrong Routes           *
* 10.List Those Excluded For Medical Reasons        *
*---------------------------------------------------*
* 11.Quit                                           *
*****************************************************
7
```

Comp-Num:	Started?	Course:	Last Rec Time:	Last Seen:	Track?:	Finished?
1	yes	D	13:20	9	0	yes
2	yes	C	12:30	1, CP	0	yes
3	yes	B	12:55	3, MC	0	no
4	no	D				
5	yes	D	14:19	9	0	yes
6	yes	C	12:45	1, CP	0	yes
7	no	B				

```
Type 'c' to continue...
```

(d) Printout of log file

In the final section of my program evidence, I will now show the log file produced after using the event manager. Tis will display all the actions I, the user chose to perform while using the system. As you can see, the log file includes a separator where the program is opened and closed, and for each log has a specific time and date of the function.

In “log.txt”:

```
----------New Occasion Of Opening System---------- - Sun Mar 17 14:13:26 2013
Started Event Manager Program - Sun Mar 17 14:13:26 2013
Entered Checkpoint File Name - Sun Mar 17 14:13:52 2013
Supplied Checkpoints File - Sun Mar 17 14:13:57 2013
Queried The Location A Competitor - Sun Mar 17 14:14:02 2013
Queried The Location A Competitor - Sun Mar 17 14:14:07 2013
Queried Competitors Not Started - Sun Mar 17 14:14:10 2013
Queried Competitors On Course - Sun Mar 17 14:14:11 2013
Queried Number Of Competitors Finished - Sun Mar 17 14:14:14 2013
Entered Checkpoint Number Manually To Be Updated - Sun Mar 17 14:14:21 2013
```

Entered Current Checkpoint Time Manually - Sun Mar 17 14:14:33 2013
Supplied Manually A Checkpoint Time - Sun Mar 17 14:14:33 2013
Inserted Medical Checkpoint Update For Entrant - Sun Mar 17 14:15:49 2013
Entered Medical Checkpoint Number - Sun Mar 17 14:15:57 2013
Entered Departed Or Arrived Status Of Checkpoint - Sun Mar 17 14:16:00 2013
Entered Medical Checkpoint Time - Sun Mar 17 14:16:00 2013
Inserted Medical Checkpoint Update For Entrant - Sun Mar 17 14:16:04 2013
Queried Competitors Excluded Competitors - Sun Mar 17 14:16:09 2013
Queried Competitors Excluded At A Medical Checkpoint - Sun Mar 17 14:16:15 2013
Queried Competitors Excluded At A Medical Checkpoint - Sun Mar 17 14:22:22 2013
Closed The System - Sun Mar 17 14:22:27 2013
----------New Occasion Of Opening System---------- - Sun Mar 17 14:29:44 2013
Started Event Manager Program - Sun Mar 17 14:29:44 2013
Entered Checkpoint File Name - Sun Mar 17 14:30:08 2013
Entered Checkpoint File Name Incorrectly - Sun Mar 17 14:30:11 2013
Entered Checkpoint File Name - Sun Mar 17 14:30:11 2013
Supplied Checkpoints File - Sun Mar 17 14:30:19 2013
Printed A Results Table - Sun Mar 17 14:30:26 2013
Closed The System - Sun Mar 17 14:37:54 2013

4. Program Descriptions

For this, the final section of the report, I will now give a brief description on each of the programs I implemented. I will give an overview on what each one does, looking at the functionality that was required on the brief, and then looking at the functionality of what I have achieved. I will also discuss for each one about any issues I had when creating them, and how I could have improved upon these. A key part of the discussions for each program will be about the assumptions I made before implementing which affected the features of each function.

Event Creator

The first program I chose to develop was the first program is the brief. Required of me here was simply to allow users to create a range of text files or data files containing an events name, date, time, entrant information and courses. Because I used C++, it meant I could take advantage of the Strings, which made input and output much simpler. When it came to the date and time inputs for the event, I made it so the system only asked for a string collection of the required format, which meant flexibility in inputs such as the date, in case the user wanted to enter it in a different format. This was acceptable, because I knew the event manager would only be reading in the date as a collection of chars anyway. I assumed in this program that there would be only ever one event at a time during the creation of a program, hence it overwriting the name.txt file. If it was required that there be more possible events going on at the same time, I could easily provide an option for this where the user would enter the required output name for the text file for each different event. Alternatives could be to create a different directory for each event and then place the files in.

The programs' section feature was the input of the entrants. Again to make it simple I used strings as inputs for the name and course. Before entering the entrant's information, the system asks how many entrants are to be entered. This stops erroneous input from the user for instance if they enter not enough/ too many entrants if they have them written down on file before putting them into the system. The final feature was similar to the entrant input by asking how many courses would be required, but also asked how many nodes would be on each course. In addition to this, it helped avoid bad nodes being entered into each course by running through a check to see if the node inputted actually existed. This was a feature enabled by reading in the nodes file before getting the courses inputs. It would simply asked for another node if the node inputted did not exist.
After each function gained the information for the specific part of the event, it stored each in structure and wrote the structure or array of structures to files.

Checkpoint Manager

In the second system I was required to implement a graphical/ command line program allowing users to insert different entrant's latest checkpoint times and append these to the checkpoint times file. I started by creating the command line interface which would accept the files required to run the system (entrants file, nodes file, courses file, and the checkpoint file to be added to). These formed the basis on what would be displayed within the GUI. I implemented a check for the files inputted ensuring that uses entered an existing file for entrant and other information and also implemented a check for file locking. When the user entered the checkpoint file they wanted to add to, I ensured that the file was not already in use, and locked it if it wasn't. This meant that if the user attempted to edit the times whilst it was already open by the same system on another section of the course that it would not allow it, therefore preventing any file corruption or loss of data.
Once the user enters the required parameters on the system, it then opens the graphical side of the application. This then takes in the entrant and nodes information and lists entrants within a combo box ready for selection by the user. The same takes place for the checkpoint selection combo box. Though this is implemented slightly different in the sense that after the nodes file is read at the start of the program it loops through these and checks whether each one is a junction or if they are time

and medical checkpoints. This means that the list of checkpoints is refined before hitting the graphical interface side, reducing the need for error checking further when the user selects a checkpoint. By the use of radio buttons, this has meant the users only have specific options of either time or medical checkpoints, and within the medical checkpoints selection, the recording can only be set to either depart or arrive. I chose to implement a JSpinner box for the time selection, as this again reduced the need for error checking on input, and also meant users could input the time with ease. After selection of the checkpoint details, the user can press the submit button, which then sends the data to another method for printing the details to the checkpoint file. Within this method, there also contains a function link which does the important procedure of checking if an entrant was/ was not expected at a checkpoint. It does this by looking at the entrant's relevant course, then sees if the users chosen checkpoint does not match any of the courses checkpoints. Finally, the user is alerted that the checkpoint has been written, and also with the addition of telling them that the entrant was not expected if that is the case.
Once the user has finished with any additions to the checkpoint file, upon closing the program a method is called to unlock the checkpoint file and release it for other programs to use the file.
I decided to create a JavaDoc using the comments of this program for a more concise and neatly organized explanation of the methods and also implemented a JAR file to be run straight from an executable file to make it easier for users.

Event Manager

In the final program asked of me to implement, this was a simple change that needed to be made to a previous assignment program, which would use the data created and changed using the two programs previously and add the extra function of logging the users actions when interacting with the system. To do this, I found a simple was by adding a function too the main.c file which would be called at any place in the program where I felt it would need to be run. The function is shown below:

The function could then be called by:

The output then ended up within a text file named 'log.txt' which was added to every time the system performed any actions thee user required. I found that because I used my previous assignment to base my event manager on, it made the work a little easier, although I still found that any problems I had in that assignment carried over into this assignment. For instance, I found that in the 'print results' function, the track number did not appear to show.
Overall in my assignment, I have learnt a good deal about the C++ language along with the file locking options available in these languages. I selected the programs to be written on the languages I chose based on what I thought would be more appropriate and easier to implement. For instance, Swing in Java for the checkpoint manager, and writing files in C++ was the more efficient methods of going about the task.