University of Technology, Jamaica
School of Computing and Information Technology
Programming II (CMP1025)
Semester 1 - 2015/2016 (Winter) EOS Student Project

Lecturers: C. Anuli, S. Braham, L. Rattray, E. Gordon

Date Given: Week #5 (Week of Sep. 21, 2015)

Due Date: Week # 13 (Week of Nov. 16, 2015)

Introduction:

This project is designed to test each student's ability to implement the concepts he/she learnt in this course, week 1 through week 11. Specifically, you will have to demonstrate the proper use of modularity, repetition, random numbers, arrays, structures and files. The grading scheme at the end of this document is designed to measure how well you have implemented each of the above concepts to solve the given problem.

Name: _____

ID#: _____

Date/Time of interview: _____

Background

In the year 2014 a covert unit called the Jamekca Major Crimes Unit (JaMCU) was launched to tackle high-level corruption and lottery scam in the parish of St. Junes. The unit was to be headed by a former high-ranking member of the Jamekca's Military Investigation Unit (JaMIU), and would also have agents from the United Kingdom (UK). The UK plans to assist the Jamaican Government with the financing of the unit. One of the functions of the unit is to track phone calls of persons suspected of being involved in corruption and lottery scam.

The unit requires application software designed to keep track of all phone calls to and from the phone numbers of suspects. The software will maintain three files: suspect phone numbers, call log, and call statistics. The **suspect phone numbers file** contains the fields *phone number*, *first name*, *last name* and *case number*. The **call log file** contains the fields *date*, *time*, *incoming/outgoing*, *phone number*, and *minutes spent on call*. The **call statistics file** contains the fields *phone number*, *number of calls in*, *number of calls out*, *total minutes on incoming calls*, *total minutes on outgoing calls*. You are tasked with designing, implementing and demonstrating a simulation of this system. The system is to perform the following tasks:

- **Initialize System** – When this menu option is selected, the application creates the three files, or erases their contents if they already existed. The suspect phone numbers file and the call log file are to be created as *plain text files*. The call statistics file is to be created as a *binary file* and store phone records in the range 5000000 and 5009999.

- **Add Suspects Phone Number** – When this menu option is selected the user is prompted to enter the suspects *telephone number* (in the range 5000000 and 5009999), *first name*, *last name* and *case number*, then this information is added to the end of the suspect phone numbers file. The file should be laid out in a neat tabular format and be able to be viewed in a text editor.
- **Simulate Call** – When this menu option is selected, the application randomly selects one of the telephone numbers from the suspect phone numbers file, and randomly decides if this is an incoming call to that number or an outgoing call from that number. It then randomly chooses a telephone number for the incoming call or outgoing call (in the range 5000000 and 5009999), which will represent the telephone number; the call is coming from or the number being called. The application then randomly generates the number of minutes spent on the incoming/outgoing call. This information (the date of the call, the time of the call, incoming/outgoing, phone number called from, phone number called to, and minutes spent on call will be displayed on the screen and also stored in the call log file. Next, the call statistics file is updated by locating the suspect's phone number in the call statistics file, then if this was an incoming call to the suspect's number, the number of calls in is incremented and the minutes spent on call is added to the total minutes on incoming calls. If instead this was a call from the suspect's phone number then the number of calls out is incremented and the minutes spent on call is added to the total minutes of outgoing calls.
- **Display Call Log** – When this menu option is selected, the application displays the information in the call log file in a neat tabular format. If the information is

too large to fit on one screen then the user is asked to press any key to display the rest, one screen full at a time until all the information is displayed.

- **Display Call Statistics** – When this menu option is selected, the user is prompted to ***enter or select*** any phone number from the suspect phone numbers file. If the phone number is not in the suspect phone numbers file then an appropriate error message should be displayed. If the phone number is located in the suspect phone numbers file, then the corresponding call statistics should be displayed for that phone number from the call statistics file, namely the phone number, number of calls in, number of calls out, total minutes on incoming calls, and total minutes on outgoing calls.
- **Exit** – When this menu option is selected, the application should be terminated.

Required

Place yourselves in groups of TWO (2) and work together to write the code for the application using the C programming language and make use of the techniques, data types, and programming constructs you have learned in Programming II (CMP1025) from week one (1) to week eleven (11). Specifically, your code should demonstrate use of modularity, random number generation, arrays, pointers, repetition, structures, and files.

Additional Instructions:

The entire project is to be stored on a CD, and a declaration of authorship form completed and signed by each group member and handed to the Student Affairs Office during their normal opening hours on or before the due date. Students must collect a receipt when he/she hands in their project. The completed project must run and compile

without sytntax and semantic error; each group of students must schedule a time and attend an interview with their lab tutor to receive a project grade. Bring a copy of this project sheet mark scheme and the cover sheet with you to the interview and place your name and id number in the space provided. **Late projects will not be accepted.**

Note. Projects that do not run will receive zero (0).

Marking Scheme

- ***Correct functionality (Select, View, Simulate, Exit properly (free from sematic error) ……...50%)***
- ***Implementation of solution using the C Programming Language.……..50%***
  - Use of random number generation (5%)
  - Use of arrays (5%)
  - Use of repetition (5%)
  - Use of pointers (5%)
  - Use of structures (5%)
  - Use of files (10%)
  - User Interface (5%)
  - Use of documentation (comments, indentations, etc) (5%)
  - Use of knowledge, techniques, and concepts learnt outside of the syllabus (10%)

Total ……...……100%