SENG1050 – DATA STRUCTURES

FOCUSED ASSIGNMENT 2 - SINGLY-LINKED LISTS

OVERVIEW

- Change Focused Assignment 1 to use a linked list instead of an array.

OBJECTIVES

- Create and use a singly-linked list.

ACADEMIC INTEGRITY AND LATE PENALTIES

- Link to Academic Integrity Information
- Link to Late Policy

EVALUATION

- The evaluation of this assignment will be done as detailed in the Marking lecture in Week 2 of the C course.

PREPARATION

- Make sure that Focused Assignment 1 works well.

REQUIREMENTS

Changed Requirements

- Replace the array of Focused Assignment 1 with a linked list. Use malloc() to do memory allocation. Call your linked list data structure FlightNode.
    - Instead of accepting 10 flights, accept an unlimited number of flights (it is, after all, a linked list with no inherent limit other than the amount of memory you have).
    - User input of the flights should end after the user enters a "." for either the destination or date. When that happens, stop getting input immediately (i.e. if the user enters "." for the destination, stop getting input and do not get the date) and display the flight information using printFlightInfo() as in Focused Assignment 1.

    - You can change the required functions from Focused Assignment 1 if you wish.
    - Do not put linked-list-specific insertion code in fillFlightInfo() as it will cause problems in later assignments. fillFlightInfo() should get the flight information but not store it in the linked list. Honest. It is OK to set the node's next field in this function, though.

GIT REQUIREMENTS

- Use GitHub Classroom for revision control, similar to how you were required to in Focused Assignment 1. It is expected that you have a reasonable number of commits with meaningfully descriptive commit comments.

CHECKLIST REQUIREMENTS

- Create a requirements checklist. This should contain the specific requirements from this assignment as well as any relevant requirements that have been covered in lecture or that are found in the SET Coding Standards or SET Submission Standards. Do it in whatever form you wish. Hand in your completed checklist in PDF form as checklist.pdf. Not having this checklist will result in a cap of 80 on your mark.

FILE NAMING REQUIREMENTS

- You must call your source file f2.cpp.
- You must call your checklist checklist.pdf.

SUBMISSION REQUIREMENTS

- Do not hand in any other files.
- Submit your files to the *DS: Focused Assignment 2* Assignment Submission Folder.
- Once you have submitted your file, make sure that you've received the eConestoga e-mail confirming your submission. Do not submit that e-mail (simply keep it for your own records until you get your mark).

HINTS

- Please make sure that you take the destination and date in from the user **in that order** and **on separate lines**. Always take destination first and then date. This will be very significant later in Major Assignment 1.
    - Likewise, print destination before date when displaying the information at the end.
- I **highly recommend** that you use two techniques that were introduced in the C course:

- Source-level debugging will help you determine if you are creating your linked list properly. Set breakpoints and expand the *next* pointers of each node until you reach the end of the list.
- A pad of paper and a pencil will allow you to visualize how the links are happening. It will also allow you to write down the values of pointer variables, since it's very difficult to keep track of in your mind and you want to see when they change and don't change.