FOCUSED ASSIGNMENT 1 – DYNAMIC MEMORY ALLOCATION

# GENERAL COURSE OBJECTIVES

* Write a program that takes in information about flights and stores the destination and  
  date in an array of structs to be displayed. Dynamic memory allocation is used to store  
  the strings in the struct.
* Use dynamic memory allocation.
* Use structs.

# REQUIREMENTS

* Use GitHub Classroom for revision control for your source code file.
* **Input Requirements**
  + Write a program that takes in information about flights and stores the destination and date in an array of structs to be displayed. Dynamic memory allocation is used to store the strings in the struct.
  + As always, prompt the user before getting the input. It is adequate (but not necessary) to prompt once for all ten pairs, as long as you are explicit about what the user should enter.
  + The two strings must be entered on two separate lines.  
    o Assume the input string will be less than 30 characters long.
* **Struct Requirements**
  + Define a struct called FlightInfo that has two fields:
    - a pointer to a char (i.e. a null-terminated string) for the destination  
      o a pointer to a char (i.e. a null-terminated string) for the date

- Both pointers must be set up by using malloc() once you know how long each string is failing to use malloc will result in a mark of 0 and no further feedback). Don’t forget to add one to the length for the null-termination.  
• There is no array in the struct.  
• Define the FlightInfo struct before any of your functions. This does not violate the banning of global variables since these are data types, not variables.

--------------------------------------------------------------------------------------------------------------

* **Function Creation Requirements**
  + In main(), declare an array variable of struct FlightInfo of size 10.
  + Write a function called fillFlightInfo that takes a pointer to a struct FlightInfo (which is a pointer to an element of the array) and the two flight information strings containing the destination and date information as parameters and fills in the struct's fields. Note the data type of the first parameter. You are taking in a pointer to a struct, not an array and not a struct.
    - This function is also where you must allocate the two blocks of memory to contain the destination string and date string.
  + Write a function called printFlightInfo that takes the array of structs as a parameter and prints all of the information contained within the array in a nicely-formatted fashion, one flight per line.
    - The destination must be displayed in the first 35 characters of the line (left- justified) and the date must be displayed in the next 35 characters of the line (left-justified).  
      o You must use printf() width specifiers (or research how to specify widths using cout) to help with your formatting.
  + Note that fillFlightInfo must be called once for each flight (destination/date pair) while printFlightInfo must be called only once in total. You will likely want to call them from main().

-------------------------------------------------------------------------------------------------------------

* **OTHER REQUIREMENTS:** 
  + You must free all memory that you allocated using malloc(). This must happen at the  
    end of main().
  + Do not use calloc(). Do not use the C++ new operator. Do not use C++ string class objects (since they won't work properly with malloc()).
  + Do not have any other input or output.
  + Call your source file f1.cpp.
  + Call your checklist checklist.pdf.