# Program 1: C++ vs C : Storing a Linked List of Strings

# System Programming, akk5

## Objectives

- To highlight the differences in C and C++
- To familiarize the students with C programming
- To review programming practices and data structures

## Instructions

Every program we will write in System Programming will be in C; since most of you have experience with C++ it will help to write a program in both languages to allow you to compare and contrast the languages. Using the videos in Canvas as a starting point and using online references such as cppreference.com, complete the following task in both C and C++.

Write a program that reads an unspecified number of strings and displays them in the reverse order they were entered.  Your program should follow an outline similar to the following:

1. Inform the user that the program will read strings until "STOP" is entered
2. Loop
    a. Prompt the user for a string
    b. If the string is "STOP" break the loop, otherwise store it in an appropriate data structure
3. Inform the user that the data they will now be displayed in reverse
4. Display all the strings in reverse.
5. Free the space used by the data structure to store the strings

## Guidance

Make certain to watch the videos detailing some of the differences in C and C++; they don't cover them all, but the ones listed certainly cause problems for students not familiar with both languages.

As a recap from the videos:

C++ uses << and >> for I/O, these operations are defined in iostream
C++ supports classes
C++ uses new and delete
C++ has a string type defined in string
C++ supports pass by reference

C uses printf and scanf for I/O, these are defined in stdio.h
C doesn't support classes. C does have a mechanism for associating different types of data together in one type known as a struct
C uses malloc and free defined in malloc.h
C doesn't support a string class, instead it has a series of string functions defined in string.h
C only supports pass by value, you must use pointers as parameters to simulate pass by reference

# Grading Breakdown

| Point Breakdown | |
|---|---|
| *Structure (C / C++)* | (8 pts / 8 pts) |
| The program has a header comment with the required information. | (2 pts / 2 pts) |
| The overall readability of the program. | (2 pts / 2 pts) |
| Program uses separate files for main and class definitions | (2 pts / 2 pts) |
| Program includes meaningful comments | (2 pts / 2 pts) |
| *Syntax (C / C++)* | (16 pts / 16 pts) |
| Uses an appropriate data structure to store the strings | (8 pts / 8 pts) |
| Uses the same data structure in both programs to store the strings | (8 pts / 8 pts) |
| *Behavior (C / C++)* | (26 pts / 26pts) |
| Program does each of following | |
| • Reads and stores a string | 6 pts |
| • Reads and stores multiple strings | 7 pts |
| • Stops reading strings when "STOP" is entered | 6 pts |
| • Outputs the stored strings in reverse | 7 pts |
| *Total Possible Points* | **100pts** |
| | |
| Penalties | |
| Program does NOT compile | -100 |
| Late up to 72 hrs | -10 per day |
| Late more than 72 hrs | -100 |

# Header Comment

At the top of each program, type in the following comment:

/*

Student Name: <student name>

Student NetID: <student NetID>

Compiler Used: <Visual Studio, GCC, etc.>

Program Description:

<Write a short description of the program.>

*/


Example:

/*

Student Name: John Smith

Student NetID: jjjs123

Compiler Used: Eclipse using MinGW

Program Description:

This program prints lots and lots of strings!!

*/


# Assignment Information

Due Date: 8/29/2021

Files Expected:

   The following files are based on an implementation which uses a doubly linked list to store the strings.

   C++: main.cpp, list.cpp, list.hpp

   C: main.c, list.c, list.h