**CECS 282-01 (Monday) – Fall 2020**

**Advanced C++**

**Program 3 – Structs and Pointers**

**Due: October 12, 2020**

Create student structure with the following fields:

* Name (cstring or null-terminated character array)
* Student ID (int – unique random value between 1000 and 9999)
* grade (char – Values A thru F)
* birthday (myDate – random value: range 1/1/2000 to 12/31/2005)
* Home Town (string)

Create an array of pointers to students of size 10. Example: Student \*stuPtr[10];

Write a function that populates the array with 10 students. Example: populate(stuPtr);

Write a display function that displays the contents of the array on the screen as shown below – nicely formatted and left justified.

The displayed list should be nicely formatted with column names like this: All columns should be left-justified.

Name Student ID Grade Birthday Home Town

Tom Thumb 1002 C January 1, 2002 Small Ville

Fred Flintstone 1995 D February 3, 2003 Bedrock

Sponge Bob 2987 B June 3, 2001 Bikini Bottom

Create a menu that shows the following options:

1. Display list sorted by Name
2. Display list sorted by Student ID
3. Display list sorted by Grade
4. Display list sorted by Birthday
5. Display list sorted by Home Town
6. Exit

You need to write a sorting function for each of the menu items – 5 options needs 5 functions.

Note:

You must create a function that returns a date between a range of 2 dates.

You will use the myDate class in this program – you will not create any other class. The Student structure is NOT a class.

Take advantage of your myDate class that you just wrote. Also, it might be helpful to create a new function that returns a string for the date format:

string myDate::toString( );

To help with formatting, you may want to use the library <iomanip> which includes the setw()

**Prog#3 Teaching Objectives**

* Intro to pointers
* Increment pointers
* Intro to structs
  + Structs are data focused – typically do not have constructors
  + Classes are object focused
* Introduce cstrings – null-terminated character array
* Compare and contrast string with cstring
* Manage cstrings
  + strlen, strcpy, strcat, strcmp,
* Create and use simple sorting algorithm (bubblesort)
* Array names act like pointers (const pointers)
* sizeof( )
* Use <iomanip> for format things nicely
* Composition – including the myDate object inside the student structure – student “has-a” myDate

**What to submit on Demo day:**

**Submit 1 file which will contain ALL of your code: your main program, the Student struct and the myDate class.**

**Screenshot of the program demo. Select option 1 and then select option 5.**