CECS 282-01 (M/W) – Spring 2021
Advanced C++
Program 3 – Structs and Pointers
Due: March 8, 2021

Create student structure with the following fields:

- Name (cstring or null-terminated character array)
- Student ID (int – unique random value between 9000 and 9999)
- grade (char – Values A thru F)
- birthday (myDate – random value: range 1/1/1995 to 12/31/2005)
- Home Town (string)

In the main program, create one Student pointer
Create a function called populate that passes that pointer
In the populate function, dynamically create an array of Student pointers using the new operator, then create 10 students and populate them according to the rules listed above.

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	9002	C	January 1, 2000	Small Ville
Fred Flintstone	9995	D	February 3, 2004	Bedrock
Sponge Bob	9987	B	June 3, 2003	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() function.

This is an example of how your main function might appear:

```
int main()
{
      Student * sptr;  // declare a Student pointer
      populate(sptr);  // pass the pointer to the populate function
      display(sptr);   // display 10 students
      sortByName(sptr);     // sort the students by name
      display(sptr);   // display the students
      return 0;
}
```

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

Hi Class,

There is an important change to the current programming assignment. I have made the change to the assignment that is currently on BeachBoard.

Below is the example program from the assignment. Notice there is now an array of Student pointers in the main function as well as a pointer to a Student pointer (Student ** sptr) This should make the program much more fun...

```
int main()
{
Student * sptrArray[10];   // declare an array of Student pointers
Student ** sptr = sptrArray; // Student pointer pointer
populate(sptr); // pass the pointer to the populate function
display(sptr);  // display 10 students
sortByName(sptr);     // sort the students by name
display(sptr);  // display the students
return 0;
}
```