Tennessee TECH

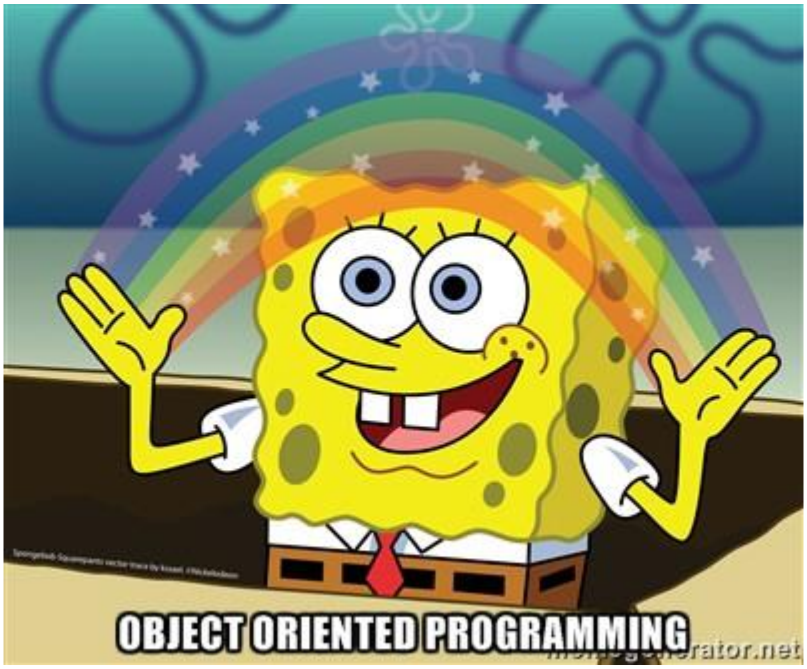
CSC1310: LAB 2

CONCEPTS

- Structures
- Classes
- Header Files
- Functions

DESCRIPTION

From Structures to Classes



In last weeks' lab, you built a structure to handle strings, called Text. But as we have been talking about in lecture, while structures are nice they have several drawbacks, including the fact that our users can directly access all of the items in our structure. So that is where classes come in. For this weeks' lab, you will be taking the structure you built last week and converting it into a class.

SPECIFICATIONS

Build a CLASS named Text to handle strings.

You should create a class specification file, named **Text.h** and a class implementation file, named **Text.cpp**.

PRIVATE VARIABLES

```
const char* textArray;   //A character string holding my "String"
int textLength;          //The size of my string
```

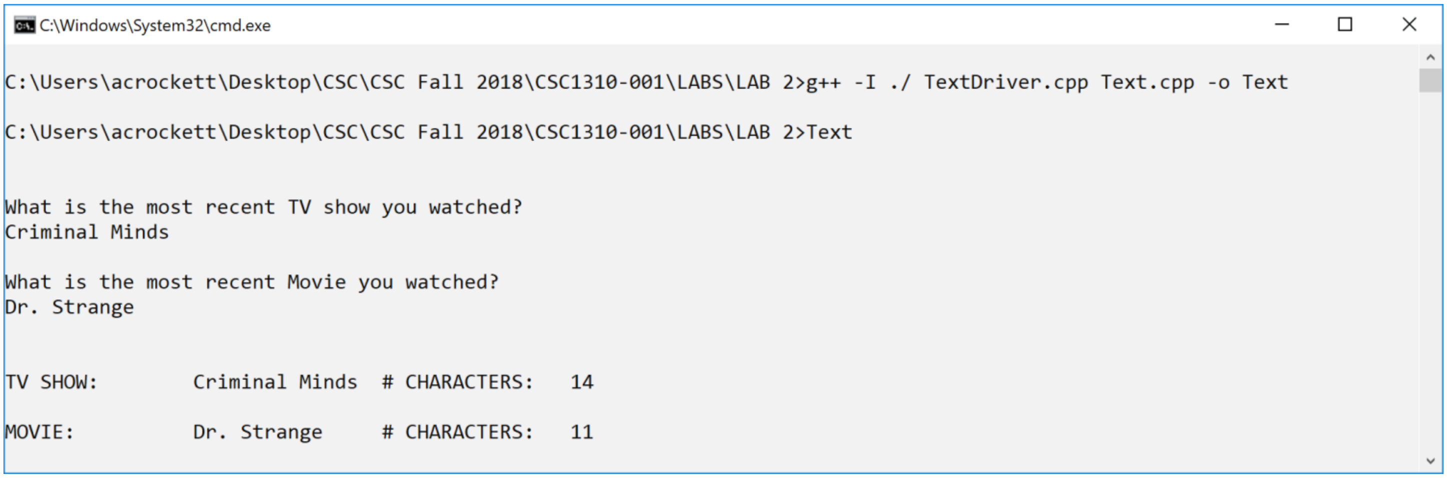
By placing these two items in the private area, we prevent the user of our class from directly manipulating any of the data.

PUBLIC FUNCTIONS

```
/*
    Function Name:  Text (constructor)
    Parameters:  Send a pointer to a constant character array or a string literal to this function
    Purpose:  called automatically when Text object is created, dynamically allocates a character array which
        contains the character array passed to the function.
*/
Text(const char*);

/*
    Function Name:  ~Text (destructor)
    Purpose: release dynamically allocated memory for the c-string in the Text object
*/
~Text();

/*
    Function Name:  displayText()
    Parameters:  none
    Returns:  none (void)
    Purpose:  prints out the string (character array)
*/
void displayText() const;

/*
    Function Name:  getText() (accessor function)
    Parameters:  none
    Returns:  pointer to a constant character array
*/
const char* getText() const;

/*
    Function Name:  getLength() (accessor function)
    Parameters:  none
    Returns:  the length of the string
*/
int getLength() const;

/*
    Function Name:  editText()
    Parameters:  pointer to a constant character array
    Returns:  none
    Purpose:  This function first deletes the DMA string that it was pointing to and then
        dynamically allocates a new character array, places the c-string passed
        to this function inside this new character array and then makes the textArray pointer
        point to this new array.
*/
void editText(const char*);
```

DRIVER

Test your code with the provided driver.cpp to see if it works! You should get the same output as below!



```
C:\Users\acrockett\Desktop\CSC\CSC Fall 2018\CSC1310-001\LABS\LAB 2>g++ -I ./ TextDriver.cpp Text.cpp -o Text

C:\Users\acrockett\Desktop\CSC\CSC Fall 2018\CSC1310-001\LABS\LAB 2>Text


What is the most recent TV show you watched?
Criminal Minds

What is the most recent Movie you watched?
Dr. Strange


TV SHOW:        Criminal Minds  # CHARACTERS:   14

MOVIE:          Dr. Strange     # CHARACTERS:   11
```

WHAT TO TURN IN

Zip all the following files and upload to ilearn.

- TextDriver.cpp
- Text.h
- Text.cpp