

Thomas Gibbons

October 27, 2016

ECE 3220

LAB 8

Objective

The objective of this lab was to become familiar with C++ using classes and objects. We reviewed what we had learned about classes and derived classes. We recalled linked lists and stacks along with the push and pop methods.

Discussion

In this lab we learned to take in strings and convert them into Morse code. We created three classes. One was a message class to store the users input. The input is all done in the arguments, but for each string a message object is created. In that object the word is stored in a string english and it is placed there by the parametric constructor.

```
class message{
private:
protected:
    string english;
public:
    message() ;
    message(string) ;
    ~message() ;
    virtual void printInfo() ;
};

class morseCodeMessage:public message{
private:
    string *morse;
    void translate();
    int index;
public:
    morseCodeMessage() ;
    morseCodeMessage(string) ;
    ~morseCodeMessage() ;
    morseCodeMessage* next;
    void printInfo() ;    //new standard
};
```

The second class was for the Morse code message and was derived from the original message class. When it is defined, it also takes the user input and defines a message object within itself because of the derivation. The parametric constructor for this class takes the input and calls the translate method and converts it to Morse code instantly. Because it is a derived class and they both have a printInfo method, the morseCodeMessage class has an override method to print.

The third class is for a stack of objects specifically morseCodeMessage objects. It keeps track of where the start of the stack is and how many objects are on the stack. Its constructors initialize the stack with either no objects or the first object of the stack. Its methods will add or subtract from the stack with the push and pop. And the print stack method uses the pop to provide a message and use that message objects print method to print and continues for each message.

```
class messageStack{
private:
    morseCodeMessage* stack_top;
    int numObjects;
public:
    messageStack();
    messageStack(morseCodeMessage current_obj);
    void push(morseCodeMessage* current_obj);
    morseCodeMessage pop();
    void printStack();
};
```

The logic behind the main is fairly simple as it first tests to see if there are argument messages to work with. Otherwise it will display error. Next it will check how many arguments there are and creates an array of object pointers. This was done because I did not know how to create an array of objects and parametrize each of the objects. After creating the array of pointers I then set up a loop to parametrize each member of the array of pointers. Along with that I pushed each member to the stack during the loop. I then printed the stack. Then I created another loop in which I deleted all of the objects I allocated along with deleting the array of pointers in which I allocated.

Some example outputs are as follows.

```
C:\Users\Thomas\Desktop\C code\LAB8>lab8 I am getting a good grade on this lab
i = ..
am = .- --
getting = --. . - - .. -. ---.
a = .-
good = --. --- --- -..
grade = --. .-. .- -.. .
on = --- -.
this = - .... .. ...
lab = .-.. .- -....

C:\Users\Thomas\Desktop\C code\LAB8>
```

```
C:\Users\Thomas\Desktop\C code\LAB8>lab8 I have successfully allocated and deallocated all the memory I need  
i = ..  
have = .... -.- ...- .  
successfully = ... ..- -..- -..- . ... ..- -..- -..- -..- -..- -..- -..-  
allocated = -. .... -- -..- .- - - -..  
and = -. -.- -..  
deallocated = -.. . -.- -..- -..- -..- -..- -..  
all = -. .... -..  
the = - .... .  
memory = -- . -- --- .-- -..-  
i = ..  
need = -. . . -..  
  
C:\Users\Thomas\Desktop\C code\LAB8>
```

As always a link to my github account is <https://github.com/tgibbons95/Lab8>.

Commits on Nov 1, 2016



Final Version v2

tgibbons95 committed 28 minutes ago



4a33e38



Final Version

tgibbons95 committed 32 minutes ago



7598bb4



2 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



tgibbons95 Final Version v2

Latest commit 4a33e38 29 minutes ago



lab8.cpp

Final Version v2

29 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README