

HW2 – Constructing a Class

Overview

This homework assignment's purpose is to get you familiar with constructing classes in C++. Project data are included for the CLion IDE. Tests for the project are also included and can be run from within the development environment. Please note: you must split the declaration and definition of the class into the header file and source file, respectively, so that tests may include the header file by itself. Files to be submitted are as follows:

NumericalMenu Class: **NumericalMenu.cpp**
 NumericalMenu.h

The class *must* be in the files specified above and follow the specification in this document. If you use Eclipse, you will need to create a project for the assignment.

Objectives

Upon completion of this activity, students should be able to...

- Design, declare, and define a simple C++ class
- Implement C++ class methods and attributes (also known as member functions and variables)
- Match a class specification so that code can be imported into another program

Instructions

You will construct a class for the creation and management of menus for console-based programs, according to the specifications outlined in this document.

WARNING: Your output must match the examples **EXACTLY to receive credit**. Deviations will result in reduction of score on the assignment. Thoroughly test your code to be sure it compiles and executes properly according to standard C++. **Submissions that do not compile will be marked zero.**

NumericalMenu

The class *NumericalMenu* should be defined in *NumericalMenu.h* and contain the following methods:

```
// Default constructor
```

```
NumericalMenu()
```

Default constructor for this class.

```
int addOption(string option)
```

Add the passed in **option** to the menu. Returns the number assigned to the new option in the menu.

```
void setCancelText (string cancelText)
```

Set the text displayed for cancellation to **cancelText**. (Defaults to *"Cancel"* if not called.)

```
void setErrorText (string errorText)
```

Set the text displayed on receipt of invalid input to **errorText**. (Defaults to *"Error!"* if not called.)

```
void setPrompt(string prompt)
```

Set the initial **prompt** that precedes the menu options. (Defaults to *"Choose an option:"*)

```
void setRepeatPromptOnError(bool shouldRepeat)
```

Set whether or not the prompt is repeated on receipt of invalid input; prompt will be repeated if **shouldRepeat** is *true*. (Defaults to *true* if not called.)

```
int run() const
```

Runs the menu; does not return until the user provides valid input. Returns *-1* on cancellation, and the option number otherwise.

```
int size() const
```

Returns the current number of menu options in this menu.

You may, at our option, add any *private* class methods that help you accomplish this goal. DO NOT add any additional *public* methods or variables, as this violates encapsulation safety.

Here is an example of a program that might use this class:

```
#include <iostream>
#include "NumericalMenu.h"

int main()
{
    NumericalMenu menu;
    menu.setPrompt("Select an option:");
    menu.addOption("Enter new values");
    menu.addOption("Help");
    menu.addOption("Save");
    menu.setCancelText("Exit");

    int result = menu.run();
    cout << "Returned: " << result << endl;
}
```

When run, here is what the output would look like:

```
Select an option:
1 - Enter new values
2 - Help
3 - Save
4 - Exit

17↵
Error!

Select an option:
1 - Enter new values
2 - Help
3 - Save
4 - Exit

0↵
Error!

Select an option:
1 - Enter new values
2 - Help
3 - Save
4 - Exit

4↵
Returned: -1
```

Submission

Your code and any accompanying documentation should be submitted online through the Canvas website's Assignment tab for this homework. All submission data must be submitted within the **root** of the zip file. ***You must include the source (cpp) and header (h) file in your submission.***

Submissions that do not follow this specification may be marked zero (no credit).