Lab 13

CSE 165: Object Oriented Programming

Spring 2022

100 points

1. Dynamic Arrays (30 Points)

In the `dynArray.cpp` file, a template class *DynArray* is needed for it to correctly compile and display the given output when executed. Namely, *DynArray* can be instantiated with an arbitrary type. Provide its implementation in a header file `DynArray.h`

Expected Output from `dynArray.cpp` :

```
2 1.0806 -0.832294 -1.97998 -1.30729
2 1 0 -1 -1
```

Each line outputs the contents of the array separated by a space. The array is populated from results of the cosine function for inputs `[0, 1, 2, 3, 4]` , where the values are casted to either a `float` or `int` type.

2. Iterators (30 points)

This exercise demonstrates the concept of an iterator. Extend your *DynArray* class of the previous exercise so the `iterators.cpp` file works correctly. Your implementation should allow the for loop to function using the *Iterator* class and the *DynArray* class should define element access (hint: operator overload).

Expected output from `iterators.cpp` :

```
0 : 2
1 : 1
2 : 0
3 : -1
0 : 2
1 : 1.0806
2 : -0.832294
3 : -1.97998
```

3. STL Map (40 Points)

Given a text file `file.txt`, print all the words with their corresponding count. You need to use STL's *unordered_map* and `simplemap.cpp` file to achieve this task. The map should be organized so that words are the keys, and the frequency of the word occurence is the value, like so: `unordered_map<string, int> wordFreq`. Refer to `wordcount.txt` for the expected output.