Assign5

Expéditeur : Ndjeuha gihane (ndjeuha.gihane@yahoo.com)

À : bonspi2000@yahoo.fr

Date : mardi 10 novembre 2020 à 05:56 UTC−8

CS300assignment5

Due Date: November 20, 2020 11:59pm

A Simple Barcode Reader Application

In this assignment, you are asked to design and develop a Barcode reader application. You are provided with an input file with items with UPC code and description. The Universal Product Code (UPC) is a barcode symbology that is widely used for tracking trade items in stores. UPC consists of numeric digits, that are uniquely assigned to each trade item. The data is obtained from http://www.grocery.com/open-grocery-database-project/.

You are going to read the file into binary search tree and allow user to search by a barcode.

Here are the functional requirements of the application:

- read file into binary search tree: parse the file content and store as product object in a binary search tree. Key is UPC code and Value is the description of the item
- search by UPC code: application takes the UPC code as an input and prints the description of the product.
- print the total time to complete the search
- handle errors for invalid input, file not found

Sample runs are provided below:

Sample Run-1:

```
UPC Code: 657622604842
Honest Tea Peach White Tea
Lookup time: 1 miliseconds
```

Sample Run-2:

```
UPC Code: 071072030035
Coffee Espresso Decaf
Lookup time: 10 miliseconds
```

Sample Run-3:

> UPC Code: 1111
> Not found

You are provided with a starter code (`ass.cpp`). The class is partially implemented, and your task is to implement the rest of it.

> A sample code to measure the execution time in C++

```
#include <iostream>
#include <time.h>

using namespace std;

int main()
{
        clock_t t;
        t = clock();

        size_t size = 100000;
        int *pInt = new int[size];         //just for testing
        for(size_t i = 0; i < size; i++)   //randomizes an array
                pInt[i] = rand();

        t = clock() - t;
        cout << "time: " << t << " miliseconds" << endl;
        cout << CLOCKS_PER_SEC << " clocks per second" << endl;
        cout << "time: " << t*1.0/CLOCKS_PER_SEC << " seconds" << endl;

        delete [] pInt;
        return 0;
}
```

How to Submit

- You’ll submit your work via GitHub Classroom assignment created by your instructor
- You’ll receive a link to your assignment via Assignment tool on Canvas
- After you accept the assignment, you’ll enter the assignment repository on GitHub
- Click “Clone or Download” and clone the assignment onto your computer
- After you make changes, commit them. Commits are essentially taking a snapshot of your projects.
- At some point you'll want to get the updated version of the assignment back onto GitHub, either so that we can help you with your code, or so that it can be graded. You can do this by performing push operation.
- If you are **late** in submitting your work, please submit your assignment via **CANVAS**.

How to Evaluate

The following rubric describes how your work will be evaluated. **Correctness** (90 points)

- [90] Program is correct in object oriented design and function; meets specification
- [75] Program output is correct but elements of specification missing, e.g. variable/method declarations.
- [45] Part of the specification has been implemented, e.g. one out of two required subprograms.
- [20] Program has elements of correct code but does not assemble/compile.

Readability (20 points)

- [10] Programmer name and assignment present. Sufficient comments to illustrate program logic. Well-chosen identifiers.
- [7] Programmer name present, most sections have comments. Fair choice of identifiers
- [5] Few comments, non-meaningful identifiers
- [0] No programmer name. No comments. Poor identifiers