CC2

Saturday, February 15, 2020 3:19 PM

CC2 - Week 5 - Spiral 2 Submission

Nolan Downey - ndowney@nd.edu

Changes from Spiral 1 to Spiral 2 are marked in red. Everything I found in common between Professor Morrison's solution and mine stayed in black, and unnecessary elements were removed

Problem 3 - Nolan Downey

1) Identify Objectives
    a. Create a class which contains the four different elements
        i. URL - string
        ii. Website Header - string
        iii. Date Visited - string
        iv. Time Visited - string
    b. Create a main driver which reads in the information from a text file and stores it in the class
    c. Use a data structure to store the classes, then overload the ostream operator to print out the elements in the specific order required

2) Identify Risks and Alternatives
    a. Risk: Use a Hash Table to easily store the information
    b. Alternative: Have to keep the elements in the order that they are accessed
    c. Risk: Do not know how many websites are going to be entered
    d. Alternative: Store the websites into a data structure that can handle an unknown number - can't use an array
    e. Risk: Need to know whether the characters are in Unicode or ASCII
    f. Alternative: We can assume the characters will be in ASCII because they will be run on the Notre Dame computers
    g. Risk: Ensure that the right number of arguments is run on the command line
    h. Alternative: Write a function that ensures the right number of arguments is given
    i. Risk: Duplicates can be given in a browser history
    j. Alternative: Use a data structure that can take in multiple versions of similar elements

3) Product Testing and Development
    a. Use a std::vector to store the class Site
    b. Create an empty std::vector
    c. The class Site should contain a constructor and an overloaded operator to store the elements from the text file and then a private element for each data type of the elements of the browsing history
    d. As the elements are read from the data file, they are stored into the class Site, then each class is pushed into the vector using the .push_back() method
    e. The following testing method displays the logic behind the approach
    f. Also the elements will be stored in the order that they are accessed
    g. Use smart iterator to print the results in order

Vector	
0	Element 1 - Class Site - URL

	- Website Header - Date Visited - Time Visited
1	Next Element - Class Site - URL - Website Header - Date Visited - Time Visited
…7	…std::vector has initial size of 8

Problem 1 - Connor Ruff

1) Identify Objectives
    a. Create a class which contains the five different elements
        i. Song ID - unsigned int
        ii. Artist/Band Name - string
        iii. Song Title - string
        iv. Album Name - string
        v. Year Recorded - unsigned int
    b. Create a main driver which reads in the information from a text file and stores it in the class
    c. Use a data structure to store the classes, then overload the ostream operator to print out the elements in the specific order required

2) Identify Risks and Alternatives
    a. Risk: Use a Hash Table to easily store the information
    b. Alternative: Have to keep the elements in the order that they are accessed
    c. Risk: Do not know how many websites are going to be entered
    d. Alternative: Store the playlist into a data structure that can handle an unknown number - can't use an array
    e. Risk: Need to know whether the characters are in Unicode or ASCII
    f. Alternative: We can assume the characters will be in ASCII because they will be run on the Notre Dame computers
    g. Risk: Ensure that the right number of arguments is run on the command line
    h. Alternative: Write a function that ensures the right number of arguments is given
    i. Risk: Duplicates can be given in a playlist
    j. Alternative: Use a data structure that can take in multiple versions of similar elements

3) Product Testing and Development
    a. Use a std::vector to store the class Music
    b. Create an empty std::vector
    c. The class Music should contain a constructor and an overloaded operator to store the elements from the text file and then a private element for each data type of the elements of the playlist
    d. As the elements are read from the data file, they are stored into the class Music, then each class is pushed into the vector using the .push_back() method
    e. The following testing method displays the logic behind the approach
    f. Also the elements will be stored in the order that they are accessed
    g. Use smart iterator to print the results in order

Vector	

0	Element 1 - Class Music - Song ID - Artist/Band Name - Song Title - Album Name - Year Recorded
1	Next Element - Class Music - Song ID - Artist/Band Name - Song Title - Album Name - Year Recorded
…7	…std::vector has a initial size of 8

Problem 2 - Kelly Buchanan

4) Identify Objectives
    a. Create a class which contains the four different elements
        i. Item ID - unsigned int
        ii. Manufacturer Name - string
        iii. Item Name - string
        iv. Cost - string
    b. Create a main driver which reads in the information from a text file and stores it in the class
    c. Use a data structure to store the classes, then overload the ostream operator to print out the elements in the specific order required

5) Identify Risks and Alternatives
    a. Risk: Use a Hash Table to easily store the information
    b. Alternative: Have to keep the elements in the order that they are accessed
    c. Risk: Do not know how many items are going to be entered
    d. Alternative: Store the websites into a data structure that can handle an unknown number - can't use array
    e. Risk: Need to know whether the characters are in Unicode or ASCII
    f. Alternative: We can assume the characters will be in ASCII because they will be run on the Notre Dame computers
    g. Risk: Ensure that the right number of arguments is run on the command line
    h. Alternative: Write a function that ensures the right number of arguments is given
    i. Risk: Duplicates can be given in a receipt
    j. Alternative: Use a data structure that can take in multiple versions of similar elements
    k. Risk: Can't use floats for money since precise, not accurate
    l. Alternative: Must create a Money class with long unsigned int for Dollars, and int for Cents

6) Product Testing and Development
    a. Use a std::vector to store the class Receipt
    b. Create an empty std::vector
    c. The class Receipt should contain a constructor and an overloaded operator to store the elements from the text file and then a private element for each data type of the elements of the receipt
    d. As the elements are read from the data file, they are stored into the class Receipt, then each class is pushed into the vector using the .push_back() method

e. The following testing method displays the logic behind the approach
f. Also the elements will be stored in the order that they are accessed
g. Use smart iterator to print the results in order

Vector	
0	Element 1 - Class Receipt - Item ID - Manufacturer Name - Item Name - Cost
1	Next Element - Class Receipt - Item ID - Manufacturer Name - Item Name - Cost
…7	…std::vector has initial size of 8