|  |  |  |
| --- | --- | --- |
| Write an application in C++ to read a line of text as a single string as command line argument, parse, extract, store and display the words following the specified steps: | | |
| Requirement Tag | Requirement Description | Comments |
| VSTR/01 | Read a line of text as a single string from command line |  |
| VSTR/02 | Create a class named “Wordset” with members as below vector<string>vlist;  int addmember(string s) - add given string s to vlist int deletemember(string s) – search and delete the given string s from vlist void displaylist() – iterate the vlist and display all its contents |  |
| VSTR/03 | Parse the command line arguments, extract words and store them in vlist |  |
| VSTR/04 | Prompt the user for a word to search and delete, then call deletemember if found |  |
| VSTR/05 | Display the contents of vector. |  |
| VSTR/06 | Empty the vlist and display the size of vlist |  |
| VSTR/07 | For every unque word stored in vector, get its frequency and display. | Hint: Use a map with key as string and frequency as value |

1.Read a line of text as a single string from command line.

2.Create a class named “Wordset” with members as below  
 vector <string> vlist;  
  
 int addmember(string s) - add given string s to vlist  
 int deletemember(string s) – search and delete the given string s from vlist  
 void displaylist() – iterate the vlist and display all its contents.

3. Parse the command line arguments, extract words and store them in vlist

#include<iostream>

#include<string>

#include<vector>

using namespace std;

class Wordset{

public:

void Vlist(){

vlist.push\_back("s");

for(string s:vlist)

{

cout<<s<<" "<<endl;

}

vlist.pop\_back();

for(string s:vlist)

{

cout<<s<<" "<<endl;

}

for(it=vlist.begin();it<vlist.end();++it)

cout<<endl<<\*it<<" "<<"\t";

}

private:

vector<string>vlist={"Life","lying"};

vector<string>::iterator it;

};

int main(int argc, char \*argv[])

{ for(int i=0;i<argc;i++)

{

cout<<"Element[i]:"<<argv[i]<<endl;

}

Wordset V;

V.Vlist();

return 0;

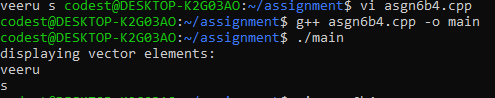
}





|  |
| --- |
| Display the contents of vector. |
| Empty the vlist and display the size of vlist |





For every unque word stored in vector, get its frequency and display.

