3460:4/521 Project 1 Report

```
Name:     Rob Bauer
UANET id: rsbauer

Generated: Wed Oct  6 13:51:47 EDT 2010

make clean
rm -f Concordance.o wordlist.o Concordance split_t.o split_t words_t.o words_t

make
g++ -c Concordance.cpp
g++ -c wordlist.cpp
g++ Concordance.o wordlist.o -o Concordance

make test
g++ -c split_t.cpp
g++ split_t.o wordlist.o -o split_t
g++ -c words_t.cpp
g++ words_t.o wordlist.o -o words_t
./split_t
blankString_resultEmptyVector_test... Passed
simpleSplit_resultVectorWithWords_test... Passed
multipleDelimiter_resultVectorWithWords_test... Passed
./words_t
blankString_resultEmptyVector_test... Passed
sortUniqueWords_resultSortedWordList_test... Passed
someDuplicateWords_resultUniqueWordList_test... Passed

./words_t
blankString_resultEmptyVector_test... Passed
sortUniqueWords_resultSortedWordList_test... Passed
someDuplicateWords_resultUniqueWordList_test... Passed

./split_t
blankString_resultEmptyVector_test... Passed
simpleSplit_resultVectorWithWords_test... Passed
multipleDelimiter_resultVectorWithWords_test... Passed

make clean
rm -f Concordance.o wordlist.o Concordance split_t.o split_t words_t.o words_t

rsbauer/Projects/Project1/Concordance/Concordance.cpp:

/*
        Concordance.cpp

        The purpose of this program is to text (through standard input) and create
        a list of sorted, unique, words in the text. You will solve this problem by
        implementing the following functions:

    * split - given text as a string and a string of splitter characters,
                returns a vector of strings in the text
    * words - given a vector of words, returns a vector of unique, sorted words


        Robert S. Bauer
        rbauer@tekro.com
*/

#include <iostream>
#include <string>
#include <vector>
#include "wordlist.hpp"

// display a token
void displayToken(const std::string &);

int main()
{
        int input;
        std::string sentence = "";
        std::cout << "Please enter a sentence for concordance calculation\n: ";

        // grab some input
        while((input = std::cin.get()) != '\n')
        {
                sentence.push_back(input);
        }

        // calculate concordance
        std::vector<std::string> tokens = words(sentence);

        // display the results
        for_each(tokens.begin(), tokens.end(), displayToken);

        return 0;
}

// Display a token
void displayToken(const std::string &token)
{
        std::cout << token << "\n";
}
rsbauer/Projects/Project1/Concordance/wordlist.hpp:

/*
        wordlist.hpp

        Declaration of word list functions

        Robert S. Bauer
        rbauer@tekro.com
*/

#ifndef INCLUDED_WORDLIST_HPP
#define INCLUDED_WORDLIST_HPP

// split string into a vector based on the chars to split on
std::vector<std::string> split(const std::string &, const std::string &);

// return a vector of unique, sorted words
// by default splits on space and .,;:!? characters
std::vector<std::string> words(const std::string &);

#endif
rsbauer/Projects/Project1/Concordance/wordlist.cpp:

/*
        wordlist.cpp

        Implement Concordance functions

        Robert S. Bauer
        rbauer@tekro.com
*/

#include <string>
#include <vector>
#include <iostream>
```

3460:4/521 Project 1 Report

```
// split words by the following characters
const std::string SPLIT_ON_CHARS = " .,;:!?";

//      Given text as a string and a string of splitter characters, returns a vector
//      of strings in the text
std::vector<std::string> split(const std::string &sentence, const std::string &splitOn
)
{
        char *ptrToken;
        char *sentence2split = strdup(sentence.c_str());
        std::vector<std::string> tokens;

        // loop through the tokens storing them and checking if out of tokens
        while((ptrToken = strtok(sentence2split, splitOn.c_str())) != NULL)
        {
                tokens.push_back(ptrToken);

                // need to reset sentence2split so the next token can be retrieved
                sentence2split = NULL;
        }

        free(sentence2split);

        return tokens;
}


// given a string, return a vector of unique, sorted words
std::vector<std::string> words(const std::string &sentence)
{
        std::vector<std::string>::iterator it;

        std::vector<std::string> words = split(sentence, SPLIT_ON_CHARS);

        std::sort(words.begin(), words.end());

        // use default comparison
        it = std::unique(words.begin(), words.end());
        words.resize(it - words.begin());

        return words;
}
```

rsbauer/Projects/Project1/Concordance/words_t.cpp:

```
/*
        split_t.cpp

        Word list method unit tests

        Robert S. Bauer
        rbauer@tekro.com
*/

#include <cassert>
#include <iostream>
#include <vector>
#include <string>
#include "wordlist.hpp"

void blankString_resultEmptyVector_test();
void someDuplicateWords_resultUniqueWordList_test();
void sortUniqueWords_resultSortedWordList_test();
void passed();

int main()
{
        blankString_resultEmptyVector_test();
        sortUniqueWords_resultSortedWordList_test();
        someDuplicateWords_resultUniqueWordList_test();
        return 0;
}

// Test passing in blank string for both the string to split and the
// string defining the characters to split on
void blankString_resultEmptyVector_test()
{
        std::cout << "blankString_resultEmptyVector_test... ";
        std::vector<std::string> tokens = words("");

        // check the size
        assert(tokens.size() == 0);
        assert(tokens.empty());

        passed();
}

// try sorting a word list that already contains unique words
void sortUniqueWords_resultSortedWordList_test()
{
        std::cout << "sortUniqueWords_resultSortedWordList_test... ";

        std::vector<std::string> tokens = words("cc bb aa");

        // check the size and contents
        assert(tokens.size() == 3);
        assert(tokens[0] == "aa");
        assert(tokens[2] == "cc");

        passed();
}

// test everything - sorting and listing only unique words
void someDuplicateWords_resultUniqueWordList_test()
{
        std::cout << "someDuplicateWords_resultUniqueWordList_test... ";

        std::vector<std::string> tokens = words("aa cc bb cc bb aa");

        // check the size and contents
        assert(tokens.size() == 3);
        assert(tokens[0] == "aa");
        assert(tokens[2] == "cc");

        passed();
}

// let the user know the test passed
void passed()
{
        std::cout << "Passed\n";
}
```

rsbauer/Projects/Project1/Concordance/split_t.cpp:

```
/*
        split_t.cpp

        Split method unit tests
```

3460:4/521 Project 1 Report

```
        Robert S. Bauer
        rbauer@tekro.com
*/

#include <cassert>
#include <iostream>
#include <vector>
#include <string>
#include "wordlist.hpp"

void blankString_resultEmptyVector_test();
void simpleSplit_resultVectorWithWords_test();
void multipleDelimiter_resultVectorWithWords_test();

int main()
{
        blankString_resultEmptyVector_test();
        simpleSplit_resultVectorWithWords_test();
        multipleDelimiter_resultVectorWithWords_test();
        return 0;
}

// Test passing in blank string for both the string to split and the
// string defining the characters to split on
void blankString_resultEmptyVector_test()
{
        std::cout << "blankString_resultEmptyVector_test... ";
        std::string testString = "";
        std::vector<std::string> tokens = split(testString, testString);

        assert(tokens.size() == 0);
        assert(tokens.empty());

        std::cout << "Passed\n";
}

// Perform some simple string splits using one character to split on
void simpleSplit_resultVectorWithWords_test()
{
        std::cout << "simpleSplit_resultVectorWithWords_test... ";
        std::string testString = "This is a test";
        std::string splitOn = " ";
        std::vector<std::string> tokens = split(testString, splitOn);

        // check the size and contents
        assert(tokens.size() == 4);
        assert(!tokens.empty());
        assert(tokens[0] == "This");
        assert(tokens[3] == "test");

        std::cout << "Passed\n";
}

// Test if splitting with multiple delimiters work
void multipleDelimiter_resultVectorWithWords_test()
{
        std::cout << "multipleDelimiter_resultVectorWithWords_test... ";
        std::string testString = "This,is.a:test";
        std::string splitOn = ":.,";
        std::vector<std::string> tokens = split(testString, splitOn);

        // check the size and contents
        assert(tokens.size() == 4);
        assert(!tokens.empty());
        assert(tokens[0] == "This");
        assert(tokens[3] == "test");

        std::cout << "Passed\n";
}
```

rsbauer/Projects/Project1/Concordance/Makefile:

```
##
# Project 1: Concordance
#
# Robert S Bauer
# rbauer@tekro.com
##

all : Concordance

Concordance : Concordance.o wordlist.o
        g++ Concordance.o wordlist.o -o Concordance

Concordance.o : Concordance.cpp
        g++ -c Concordance.cpp

wordlist.o : wordlist.cpp wordlist.hpp
        g++ -c wordlist.cpp

split_t : split_t.o wordlist.o
        g++ split_t.o wordlist.o -o split_t

split_t.o : wordlist.cpp wordlist.hpp split_t.cpp
        g++ -c split_t.cpp

words_t : words_t.o wordlist.o
        g++ words_t.o wordlist.o -o words_t

words_t.o : wordlist.cpp wordlist.hpp words_t.cpp
        g++ -c words_t.cpp

test : split_t words_t
        ./split_t
        ./words_t

testsplit : split_t
        ./split_t

testwords : words_t
        ./words_t

clean :
        rm -f Concordance.o wordlist.o Concordance split_t.o split_t words_t.o words_t
```

```
------------------------------------------------------------------------
r586 | rsbauer | 2010-09-23 18:47:10 -0400 (Thu, 23 Sep 2010) | 1 line

Quick fix: extracted string to const and added comments
------------------------------------------------------------------------
r582 | rsbauer | 2010-09-23 18:39:58 -0400 (Thu, 23 Sep 2010) | 1 line

Added a constant for word split chars
------------------------------------------------------------------------
r145 | rsbauer | 2010-09-12 11:32:33 -0400 (Sun, 12 Sep 2010) | 1 line

Concordance executable working. Added comments to unit tests and wordlist.cpp
------------------------------------------------------------------------
```

3460:4/521 Project 1 Report

```
r144 | rsbauer | 2010-09-12 11:05:27 -0400 (Sun, 12 Sep 2010) | 1 line

Words unit tests added. words() method working
------------------------------------------------------------------------
r143 | rsbauer | 2010-09-12 07:44:06 -0400 (Sun, 12 Sep 2010) | 1 line

Function signature changed (returning vector, not passing one in. Updated split().
------------------------------------------------------------------------
r137 | rsbauer | 2010-09-11 16:13:40 -0400 (Sat, 11 Sep 2010) | 1 line

Updated words_t.cpp so make test would compile without error
------------------------------------------------------------------------
r136 | rsbauer | 2010-09-11 16:11:01 -0400 (Sat, 11 Sep 2010) | 1 line

Refactored the comments
------------------------------------------------------------------------
r135 | rsbauer | 2010-09-11 15:56:12 -0400 (Sat, 11 Sep 2010) | 1 line

Split function working. Unit tests added. Updated makefile to rebuild if split_t.cpp i⤤
s updated.
------------------------------------------------------------------------
r134 | rsbauer | 2010-09-11 13:25:47 -0400 (Sat, 11 Sep 2010) | 1 line

Split_t tests were not working - fixed
------------------------------------------------------------------------
r133 | rsbauer | 2010-09-11 13:18:09 -0400 (Sat, 11 Sep 2010) | 1 line

Build error in split_t.cpp when refactored
------------------------------------------------------------------------
r132 | rsbauer | 2010-09-11 13:16:12 -0400 (Sat, 11 Sep 2010) | 1 line

SVN property changes
------------------------------------------------------------------------
r131 | rsbauer | 2010-09-11 13:02:59 -0400 (Sat, 11 Sep 2010) | 1 line

Split unit test setup and working.
------------------------------------------------------------------------
r130 | rsbauer | 2010-09-11 12:43:13 -0400 (Sat, 11 Sep 2010) | 1 line

Mocking slit function. Added comments to project files.
------------------------------------------------------------------------
r90 | rsbauer | 2010-09-10 07:30:20 -0400 (Fri, 10 Sep 2010) | 1 line

Make file built and simple hello world added to concordance - build works and concorda⤤
nce runs
------------------------------------------------------------------------
r30 | collard | 2010-09-07 18:11:34 -0400 (Tue, 07 Sep 2010) | 1 line

Setup of accounts for Project 1
------------------------------------------------------------------------
```