

ENGR1200U Introduction to Programming for Engineers (Winter 2013)

Assignment 3: Working with File I/O Using C++ (20 marks)

Due Date: By 5:30pm on Tuesday, Mar 19, 2013

The purpose of this assignment is to design and develop solutions for two file-based problems: spell checking, and encrypting/decrypting text. Submit a hard copy of all your source code as per the submission instructions section below. You are required to answer all questions. Write a separate C++ program to solve each of the two problems. Submit a hard copy of all your source code as per the submission instructions section below. **This is an individual work assignment.**

IMPORTANT NOTE: No advanced features (e.g. user functions, arrays, etc) that haven't been covered in class yet are allowed to be used in your programs. Students with solutions that use advanced features will lose a significant number of marks (if not all) for that solution.

1. *[Spell Checker - 8 marks]* The first version of the Unix-based spell checker, spell, was written in 1975 by Steve Johnson. Johnson's approach was simple: read the words in a document, isolate them, sort them, and then compare the sorted list with a dictionary. The spell program works by issuing the spell command under UNIX operating system, providing one argument – the file name to be spell-checked. The spell program displays the words that were found in the document but not in the dictionary; that is, misspelled words. The spell program, unlike Microsoft's Word spell checker, doesn't prompt the user for corrections.

Write a program that implements a simple spell checker in C++. The spell checker should work as follows: prompt the user to enter a file name to spell check, and then the program will parse the file and check if there are any tokens that are not in the dictionary. Any words that are not found in the dictionary will be printed on the screen. Note that you do not need to worry about sorting words or prompting the user for corrections.

You must test your program with the files **“bonk.txt”** (file to be spell checked) and **“dictionary.txt”** (an optimized dictionary). Both of these files can be downloaded from Blackboard (Content → Assignments → Assignment 3).

Here is an example:

```
Enter the name of the file to spell check: bonk.txt

Duke
Bonk
semi
affectionately
Bonkistry
etc
Chem
UVirginia
hangovers
everything
Duke
Bonk
UVa
Bonk
Bonk
Unprepared
```

In the above example, the user enters the name of the file to spell check (bonk.txt) and the program displays the list of words that are misspelled.

2. **[Encrypting/Decrypting Files - 12 marks]** Encryption and Decryption are two cryptographic techniques. Encryption is used to transform text to meaningless characters, and decryption is used to transform meaningless characters into meaningful text. The algorithm that does the encryption is called a cipher. A simple encryption algorithm is Caesar cipher, which works as follows: replace each clear text letter by a letter chosen to be n places later in the alphabet. The number of places, n , is called the cipher key. For example, if the cipher key is 3, the clear text “HELLO THERE” becomes “KHOOR WKHUH”.

Here, we restrict ourselves to encrypting/decrypting uppercase alphabetic characters. As you know from the ASCII table, the set of characters 'A' to 'Z' correspond to the integers 65 to 90.

Hint: The following formula can be used to encrypt a character C using the Caesar cipher algorithm:

$$E = (C - 'A' + k) \% 26 + 'A' \text{ [where } k \text{ is the cipher key]}$$

You need to figure out the formula for decrypting text on your own. And, please do not share it with other students or post it on Piazza.

NOTE: Use **cipher key = 5**.

Write a C++ program for encrypting and decrypting files. Since this program performs two different functionalities (encryption and decryption), prompt the user to select the type of cryptographic technique as shown below:

```
Welcome to ENGR1200U Cryptographic Techniques Program
Please enter your selection:
1. Encrypt
2. Decrypt
```

When the user selects 1 or 2, s/he will be asked to specify the input and output files.

NOTE: you must use the file "hello.txt" to test your encryption/decryption program. The file can be downloaded from Blackboard (Content → Assignments → Assignment 3).

Here is an example:

Assume we have the file "hello.txt" with the following content:

```
HELLO THERE
HOW ARE YOU DOING?
BYE BYE;=FOR NOW
OK; BYE!
```

If the user selects to encrypt the file (i.e. option 1), then the program will encrypt the original text in “hello.txt” and outputs the encrypted text to a new file (i.e. “hello.out”). The encrypted file contains the encrypted text that looks as follows:

```
MJQQT YMJWJ
MTB FWJ DTZ ITNSL?
GDJ GDJ;=KTW STB
TP; GDJ!
```

If the user selects to decrypt the file (i.e. option 2), then the program will decrypt the encrypted text (i.e. in “hello.out”) and outputs the decrypted text to a new file (i.e. “new_hello.txt”). The program should give the user the option whether to continue with encrypting/decrypting files (i.e. entering letter ‘C’) or exit the program (i.e. entering the letter ‘E’).

Submission Instructions

Please follow the instructions below carefully as the submission process consists of two important steps (pay attention to the grading guidelines section):

1) Submit Printed Hard Copy:

By **5:30pm on Tuesday, March 19, 2013**, you must submit a stapled hard copy of following (in order):

- Coversheet (see below) → First Page
- C++ Program source code for each problem
- Grade Sheet (see below) → Last Page

Please ensure that all the required details are provided on the cover sheet and the grade sheet including (fill in areas where you see red arrow): your name, student ID, lecture section number, and tutorial session number. It is important to include your tutorial session number on the left hand corner on both sheets.

Submission Location: On the attached coversheet, locate your tutorial session and determine the drop box number. Drop a hard copy of the assignment by the above due date in the appropriate drop box. Drop boxes are located at the first floor in the FEAS Engineering building (ENGR), near the elevators.

2) Attend Tutorial:

Attend your tutorial session (Wednesday Mar 20 / Thursday Mar 21) to demo your solutions to the TA. During the demo, the student will be tested on concepts related to the assignment.

General Guidelines

- 1) In an effort to help prevent plagiarism, assignment source code will be subject to textual similarity review. If a student submits a source code that is deemed similar in some form to one or more other submissions will automatically result in a zero grade for the assignment.
- 2) Formatting and indenting your source code is important. Follow the programming style presented during lectures and described in the textbook.
- 3) If any of your programs does not compile or run, and you are aware that it has some bugs, document this fact at the top of the source code.

Grading Penalties

- 1) -5% for not ordering sections properly (Not Organized)
- 2) -5% for not using the cover sheet as first page (No Cover Sheet)
- 3) -5% for not including grade sheet as final page (No Grade Sheet)
- 4) -50% for not attending and presenting demo during assigned tutorial date (No Demo)
- 5) -100% for no printed hard copy of assignment (No Submission)

NO LATE ASSIGNMENTS WILL BE ACCEPTED NO MATTER WHAT IS THE REASON.

You are given more than enough time to finish the assignment well before the due date.

Grading Guidelines

Criteria	Marks	Description
User Friendliness	2.0	Program description message Program output is presented on screen neatly
Program Source Code	4.0	Program is properly documented Meaningful variable names, properly formatted source code Appropriate control structures have been applied
Testing & Correctness	10.0	Programs are complete and implemented correctly
Questions	4.0	Questions related to program source code / implementation
Total	20 marks	

Write your Tutorial # here



ENGR 1200U: Introduction to Programming for Engineers

Assignment 3 Coversheet

Name			
Student ID			
Lecture Section Number (check appropriate section)	<input type="checkbox"/>	70607	Dr. Q. Mahmoud (Tues & Thurs)
	<input type="checkbox"/>	70610	Dr. E. Al-Masri (Mon & Wed)
	<input type="checkbox"/>	71041	Dr. E. Al-Masri (Tues & Thurs)

Drop Box #	Tutorial #	CRN	Teaching Assistant	Location	Day	Time
13	01	72796	Krupa Kuriakose	UA2230	Wednesdays	1:10 pm – 3:00 pm
13	02	70621	Krupa Kuriakose	UA2140	Thursdays	8:10 am – 10:00 am
13	03	70620	Krupa Kuriakose	UA2230	Thursdays	7:10 pm – 9:00 pm
14	04	72144	Musharaf Rabbani	UA2230	Thursdays	8:10 am – 10:00 am
14	05	72795	Tina Mirfakhraie	UA2220	Wednesdays	1:10 pm – 3:00 pm
14	06	71087	Tina Mirfakhraie	UA2220	Thursdays	9:10 am – 11:00 am
15	08	70617	Titus Okathe	UA2230	Wednesdays	7:10 pm – 9:00 pm
15	07	71159	Titus Okathe	UA2120	Thursdays	9:10 am – 11:00 am
15	09	71160	Saadia Gauhar	UA2240	Thursdays	8:10 am – 10:00 am
16	10	70615	Shukla Shivam	UA2220	Wednesdays	7:10 pm – 9:00 pm
16	11	70622	Shukla Shivam	UA2130	Thursdays	8:10 am – 10:00 am
16	12	70619	Shukla Shivam	UA2220	Thursdays	7:10 pm – 9:00 pm

Write your Tutorial # here



ENGR 1200U: Introduction to Programming for Engineers

Assignment 3 Grade Sheet

Name													
Student ID													
Teaching Assistant (To be completed by TA)	<table border="1"><tr><td><input type="checkbox"/></td><td>Krupa Kuriakose</td><td><input type="checkbox"/></td><td>Titus Okathe</td></tr><tr><td><input type="checkbox"/></td><td>Musharaf Rabbani</td><td><input type="checkbox"/></td><td>Saadia Gauhar</td></tr><tr><td><input type="checkbox"/></td><td>Tina Mirfakhraie</td><td><input type="checkbox"/></td><td>Shukla Shivam</td></tr></table>	<input type="checkbox"/>	Krupa Kuriakose	<input type="checkbox"/>	Titus Okathe	<input type="checkbox"/>	Musharaf Rabbani	<input type="checkbox"/>	Saadia Gauhar	<input type="checkbox"/>	Tina Mirfakhraie	<input type="checkbox"/>	Shukla Shivam
<input type="checkbox"/>	Krupa Kuriakose	<input type="checkbox"/>	Titus Okathe										
<input type="checkbox"/>	Musharaf Rabbani	<input type="checkbox"/>	Saadia Gauhar										
<input type="checkbox"/>	Tina Mirfakhraie	<input type="checkbox"/>	Shukla Shivam										

User Friendliness (2 marks)

1.0 mark: Program description message

1.0 mark: Program output is presented in a proper format

Program Source Code (4 marks)

2.0 marks: Program is properly documented

2.0 marks: Appropriate control structures have been applied

Testing and Correctness (10 marks)

4.0 marks: Spell check program is complete and implemented correctly

6.0 marks: Encrypt/Decrypt program is complete and implemented correctly

Questions (4 marks)

1.0 mark: Spell check program implementation

3.0 marks: Encrypt/Decrypt program implementation

Total (20 marks)

Additional Comments (To be completed by TA): → _____ ☐ No Grade

<input type="checkbox"/>	-1	Not Organized
<input type="checkbox"/>	-1	No Cover Sheet
<input type="checkbox"/>	-1	No Grade Sheet
<input type="checkbox"/>	-10	No Demo
<input type="checkbox"/>	-20	No Submission