ENGR1200U Introduction to Programming for Engineers (Winter 2013)

Assignment 4: Working with Functions and Arrays (20 marks)

Due Date: By 5:30pm on Tuesday, April 9, 2013

The purpose of this assignment is to use functions and arrays to re-implement Assignment #3. You are required to answer all questions. Write a separate C++ program to solve each of the three problems. Submit a hard copy of all your source code as per the submission instructions section below. **This is an individual work assignment.**

1. ***[Spell Checker - 8 marks]*** Re-implement your solution using functions and arrays. Your program must use the following function definitions and layout. You cannot add any more functions.

```
/************************************************************
* This function takes in a filename and spells check it      *
* @param file an input file stream (i.e. bonk.txt)           *
*************************************************************/
void spell(string filename)
{
       // function body
}

/************************************************************
 * This function takes a word from input file stream (i.e.    *
 * bonk.txt) and removes punctuation marks. Then, the function*
 * updates the word in the caller argument (if necessary)     *
 * @param word a string that contains a word to be filtered   *
 ************************************************************/
void wordFilter(string& word)
{
       // function body
}

/************************************************************
 * This function returns true if a word in input file stream  *
 * (i.e. bonk.txt) is in the dictionary, false otherwise.     *
 * @param word a string that contains a word to be checked    *
 * @return the Boolean status of the word (i.e. true or false)*
 ************************************************************/
bool inDictionary(string word, string dictionary[])
{
       // function body
}

/************************************************************
 * This is the main function of the program.                  *
 * @return a value to terminate the program successfully      *
 ************************************************************/
int main()
{
       // Prompt the user to enter a file to be spell checked
       spell(filename);
       return 0;
}
```

2. ***[Encrypting/Decrypting Files - 8 marks]*** Re-implement your solution using functions and arrays. Your program must use the following functions and layout. You cannot add any more functions.

```
/*************************************************************
 * This function displays the menu options as in the handout  *
 *************************************************************/
void displayMenu()
{      /* function body */   }

/*************************************************************
 * This function encrypts the content of infile and saves the *
 * encrypted text into outfile                                *
 * @param infile string (file that has raw text)              *
 * @param outfile string (file that will have encrypted text) *
 *************************************************************/
void encrypt(string infile, string outfile)
{      /* function body */   }

/*************************************************************
 * This function decrypts the content of infile and saves the *
 * decrypted text into outfile                                *
 * @param infile string (file that has encrypted text)        *
 * @param outfile string (file that will have decrypted text) *
 *************************************************************/
void decrypt(string infile, string outfile)
{      /* function body */   }

/*************************************************************
 * This function takes an character and a cipher key to return*
 * an encrypted character.                                    *
 * @param c is a char (the character to be encrypted)         *
 * @param key is an integer (cipher key given in the handout) *
 *************************************************************/
char ceaserCipher(char c, int key)
{      /* function body */   }

/*************************************************************
 * This function takes an encrypted character and a cipher key*
 * to return a decrypted character.                           *
 * @param c is a char (the character to be decrypted)         *
 * @param key is an integer (cipher key given in the handout) *
 *************************************************************/
char ceaserDecipher(char c, int key)
{      /* function body */   }

/*************************************************************
 * This is the main function of the program.                  *
 * @return a value to terminate the program successfully      *
 *************************************************************/
int main()
{
   displayMenu();
   // 1. Get user input (e.g. encrypt/decrypt and filenames)
   // 2. Call the appropriate function(s) to process encryption/decryption
   // 3. Then, get user input to continue or exit the program
       return 0;
}
```

3. ***[Command-line arguments - 4 marks]*** Re-implement your solution for Problem 2 above (with functions and arrays) using command-line arguments as follows:

```
FileCrypt -encrypt "hello.txt" "hello.out"
FileCrypt -decrypt "hello.out" "hello.ans"
```

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, April 9, 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 April 10/ Thursday April 11) 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	3.0	Program description message Program output is presented on screen neatly
Program Source Code	4.5	Program is properly documented Meaningful variable names, properly formatted source code Appropriate control structures have been applied
Testing & Correctness	8.0	Programs use functions and arrays as provided
Questions	4.5	Questions related to program source code / implementation
Total	**20 marks**	

Write your Tutorial # here

ENGR 1200U: Introduction to Programming for Engineers

Assignment 4 Coversheet

Name	
Student ID	
Lecture Section Number (check appropriate section)	☐ 70607 Dr. Q. Mahmoud (Tues & Thurs) ☐ 70610 Dr. E. Al-Masri (Mon & Wed) ☐ 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

UOIT
CHALLENGE INNOVATE CONNECT

ENGR 1200U: Introduction to Programming for Engineers

Assignment 4 Grade Sheet

Name	
Student ID	
Teaching Assistant (To be completed by TA)	☐ Krupa Kuriakose ☐ Titus Okathe ☐ Musharaf Rabbani ☐ Saadia Gauhar ☐ Tina Mirfakhraie ☐ Shukla Shivam

☐	Krupa Kuriakose	☐	Titus Okathe
☐	Musharaf Rabbani	☐	Saadia Gauhar
☐	Tina Mirfakhraie	☐	Shukla Shivam

User Friendliness (3 marks)

1.5 marks: Program description message
1.5 marks: Program output is presented in a proper format

Program Source Code (4.5 marks)

1.5 marks: Program is properly documented
1.5 marks: Appropriate control structures have been applied
1.5 marks: Functions and arrays used as provided

Testing and Correctness (8 marks)

3.0 marks: Spell check program is complete and implemented correctly
3.0 marks: Encrypt/Decrypt program is complete and implemented correctly
2.0 marks: Encrypt/Decrypt with command-line arguments

Questions (4.5 marks)

1.5 marks: Spell check program implementation
1.5 marks: Encrypt/Decrypt program implementation
1.5 marks: Command-line arguments

Total (20 marks)

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

☐	-1	Not Organized
☐	-1	No Cover Sheet
☐	-1	No Grade Sheet
☐	-10	No Demo
☐	-20	No Submission