

**FACULTY OF INFORMATICS**

|  |  |  |  |
| --- | --- | --- | --- |
| **SUBJECT’S INFORMATION:** | | | |
| Subject: | CSCI124 Applied Programming | | |
| Session: | July 2014 | | |
| Programme / Section: | J766SENG (SE) / J766CS53 (MGD) / J766CS42 (DSS) | | |
| Lecturer: | Ms. Siti Hawa | | |
| Coursework Type  *(tick appropriate box)* | ❑ Individual Assignment ❑ Group Assignment ❑ Project  ✓Lab Task ❑ Seminar / Tutorial Paper ❑ Others | | |
| Coursework Title: | **Lab Task 6** | Coursework Percentage: | 2% |
| **ASSESSMENT CRITERIA:** | | | |
| Correctness | All programs should produce the correct result as stated in the specification. | | |
| Coding | Programs should use appropriate control structures and data structures correctly based on what have been covered in the class and stated in the specification. Necessary input validations should be done. | | |
| Readability | Appropriate comments are included. Meaningful identifiers used. Proper indentation and line spacing used. | | |
| Well formatted output | Output should be well formatted with appropriate messages displayed. Numbers are shown with appropriate precision. | | |
| **SUBMISSION:** | | | |
| All completed work should be submitted online through Moodle before or on the due date provided.  **SUBMIT AS EARLY AS POSSIBLE. YOU CAN RE-SUBMIT LATER IF NECESSARY. ONLY THE LATEST SUBMISSION WILL BE MARKED.**  **IF YOU SUBMIT YOUR ASSIGNMENT TWICE, ONE SUBMMISSION BEFORE THE DUE DATE AND ANOTHER AFTER THE DUE DATE, THEN YOU WILL BE PENALIZED FOR LATE SUBMISSON.** | | | |
| DUE DATE: | **WEEK 8** | | |
| **PENALTIES FOR LATE SUBMISSION:** | | | |
| Penalties apply to all late work, except if student academic consideration has been granted. Late submissions will attract a penalty of 25% of the assessment mark per day including the weekend. Work more than (3) days late will be awarded a mark of zero. | | | |
| **PLAGIARISM:** | | | |
| **When you submit an assessment task, you are declaring the following**   1. It is your own work and you did not collaborate with or copy from others. 2. You have read and understand your responsibilities under the University of Wollongong's policy on plagiarism. 3. You have not plagiarised from published work (including the internet). Where you have used the work from others, you have referenced it in the text and provided a reference list at the end ot the assignment.   Plagiarism will not be tolerated. Students are responsible for submitting original work for assessment, without plagiarising or cheating, abiding by the University’s policies on Plagiarism as set out in the University Handbook under University Policy Directory and in Faculty handbooks and subject guides. | | | |

**COURSEWORK SPECIFICATION**

**OBJECTIVES:**

The aim of this lab is to get more experience with pointers and dynamic memory.

**TASKS:**

Your task this week is to create a game in which a player guesses letters to try to replicate a hidden word. When your program starts, prompt for an input file. The input file should be a text file that contains several words and where the first line in the file indicates the total number of words in the file. Sample input file can be as follows:

5

available

international

season

expression

location

Use dynamic memory allocation to create a two-dimensional array to keep all the words read from the file. You can assume that no word is longer than 20 characters long. Then randomly choose a word to become the hidden word.

Initially, display the hidden word using asterisks to replace each letter. Allow the user to guess letters to replace the asterisks in the hidden word until the user completes the entire word. If the user guesses a letter not in the hidden word, display an appropriate message. If the user guesses a letter that appears multiple times in the hidden word, make sure each correct instance is replaced. When the user completes the word, display a count of the number of guesses that were required. Allow the user to continue to play as many games as wanted and allow the user to quit playing a game at any time. Use pointer notation to scan through the hidden word while the game is played.

A typical game played can be as shown below:

Enter filename: hiddenWords.txt

Guess a letter in the secret word: \*\*\*\*\*\*

Guess a letter (\* to quit) > o

Guess a letter in the secret word: \*\*\*\*o\*

Guess a letter (\* to quit) > p

Sorry, p is not in the word

Guess a letter in the secret word: \*\*\*\*o\*

Guess a letter (\* to quit) > s

Guess a letter in the secret word: s\*\*so\*

Guess a letter (\* to quit) > a

Guess a letter in the secret word: s\*aso\*

Guess a letter (\* to quit) > n

Guess a letter in the secret word: s\*ason

Guess a letter (\* to quit) > e

Guess a letter in the secret word: season

Good Job!!!

You have guessed the word season in 6 tries.

Would you like to play the game again? (Y/N) : \_