

ASSIGNMENT 2

1. OBJECTIVE

The objective of this assignment is to practise on objects, arrays, strings and file I/O. You need to submit this assignment for grading.

2. LABORATORY

This lab is conducted in the Computing Lab 1 (N4-B1-10) in SCE.

3. EQUIPMENT

Hardware: Workstation/PC running under the *Linux* environment.

Software: text editor
Java 2 Platform

4. EXPERIMENT

You are required to write a Java application program for implementing a word dictionary. The dictionary can store up to 100 single English words. The class diagram for the class *Dictionary* is given below:

Dictionary
- wordList: String[] - len: int
+ Dictionary() + loadFromFile(): void + findWord(wd: String): int + insertWord(wd: String): void + removeWord(wd: String): void + saveToFile(): void + displayDict(): void

where

- The instance variable **wordList** is an array of Strings in main memory that stores all the words of the dictionary.
- The instance variable **len** is used to record the number of words that have been stored in the dictionary.
- The constructor **Dictionary()** constructs an object that creates the array of Strings **wordList** and initializes the instance variable **len** to zero.
- The method **loadFromFile()** loads the contents of the dictionary from a text file named as “dict.txt” into the array of Strings **wordList** for further processing. The file “dict.txt” contains the initial data and format as shown in Figure 2:
- The method **findWord()** returns the index for the memory location of the array **wordList** that contains the specified word **wd** in the dictionary. It returns -1 if the word is not in the dictionary.
- The method **insertWord()** inserts the specified word **wd** into the array **wordList**. The word should be inserted into **wordList** and sorted according to lexicographic order (i.e. ascending alphabetical order). Before insertion, the method should check whether the

specified word **wd** has already been stored or the **wordList** is full. For either case, an appropriate warning message should be printed on the screen.

- The method **removeWord()** deletes the specified word **wd** from **wordList**. Print an appropriate warning message on the screen if the word is not in **wordList**.
- The method **saveToFile()** saves the contents into the text file “dict.txt” from the array of Strings **wordList**.
- The method **displayDict()** displays the words stored in **wordList** on the screen.

```

compiler
data
design
information
introduction
java
logic
mathematics
mining
programming
retrieval
strcuture

```

Figure 2

Write the Java code for the *Dictionary* class using the attributes and methods specified in the class diagram. Design an application program *DictionaryApp* that uses the *Dictionary* class for implementing the word dictionary.

Important:

Remember to do all the programming inside the sub-directory *assign2* and name the source codes as **Dictionary.java** and **DictionaryApp.java** and name the compiled codes as **Dictionary.class** and **DictionaryApp.class**. The application program should provide a menu to the user as follows:

Commands:

- (1) Load the word dictionary from file
- (2) Save the word dictionary to file
- (3) Find word
- (4) Insert word
- (5) Remove word
- (6) Print the word dictionary
- (7) Quit

Test Data:

Test your application program with the following data:

1. Load words from file “dict.txt” given in Figure 2.
2. Print the word dictionary
3. Find word “information”
4. Remove word “information”
5. Insert word “test”
6. Print the word dictionary
7. Find word “nonexist”
8. Remove word “nonexist”
9. Insert word “retrieval”
10. Print the word dictionary
11. Save words into file
12. Quit

5. REPORT

- I. This lab is due at 5pm 31 October 2005 (Monday).
- II. For the report, one needs to submit hard copies of his/her in the following format:

For each constructor/method of the Dictionary class and the application program DictionaryApp of the assignment, you need to write:

1. Program logic and suggestion
 - Describe the logic of your program and justify why you design your program in this way. For example, if you use while loop in your assignment, state whether you have used a counter-controlled while loop in your program and explain why you think counter-controlled while loop is suitable for your program.
 - Suggest any other approaches for implementation.
2. program listings of Dictionary.java and DictionaryApp.java
3. the typescript file(s) to show the results of the testing of your programs using the test cases specified in this manual only

6. MARKING SCHEME

The marks for each part of the assignment are given as follows:

For Dictionary class:

- | | |
|--|------------|
| 1. Dictionary() + instance variables declaration | (10 marks) |
| 2. loadFromFile(): void | (10 marks) |
| 3. saveToFile(): void | (10 marks) |
| 4. findWord(wd: String): int | (10 marks) |
| 5. displayDict(): void | (10 marks) |
| 6. insertWord(wd: String): void | (15 marks) |
| 7. removeWord(wd: String): void | (15 marks) |

For DictionaryApp class:

- | | |
|------------------------|------------|
| 8. application program | (20 marks) |
|------------------------|------------|

The marking of each part of the assignment will be based on the following criteria:

1. Program logic and suggestion (20%)
2. Java programs (80%)
 - presentation of the Java codes (whether the program is properly indented);
 - documentation of codes (whether the program is well commented to aid understanding); and
 - correctness of the program.

7. REFERENCE

- [1] The textbook for CPE102/CSC102.