

# Assignment 1

Value **40%** of Coursework

**Individual work**

## Learning outcomes

*Students will be able to understand:*

- 1.1 Data structures
- 1.2 The applications of data structures
- 1.3 Object-oriented programming concepts
- 1.4 Methods for program testing

*Students will have acquired skills in:*

- 2.1 Data abstraction
- 2.2 The use of data structures
- 2.3 Programming at a more advanced level in a high-level object-oriented language
- 2.4 Program testing and documentation

*Students will have acquired skills in:*

- 3.1 Self-management
- 3.2 Learning
- 3.3 Communication
- 3.4 Problem solving
- 3.5 Information technology

## Submission requirements

The assignment submitted should be compressed into a .zip file, the following files should be contained in the compressed file:

- a *report* as a *Microsoft Word* document containing the code of all your classes.  
filename format: *student ID+CHC5223\_CW1\_Report.docx*
- a *.zip file* containing the project: the runnable *jar* file (if available) and all the program's source code (*.java*).  
filename format: *student ID+CHC5223\_CW1\_Files.zip*

## General requirements

All your programming must conform to “Java Conventions and Programming Guidelines” – see module Moodle site.

You must paste the key source code of your implementation into your report, as *text* or as screenshots.

## Introduction

The topics of this assignment are **array**, **linked list**, and **hash table**. The objective of this assignment is to develop a hash table data structure utilizing a double-linked list as the underlying mechanism.

## Requirements

### **Basic rules**

You must create one executable project after completing all tasks.

One Java class should be defined in one .java file respectively.

In the report, the source code of each task, together with the corresponding explanation, should be presented separately.

Failure to comply with these rules will result in **zero marks**.

### Task 1

You must design and implement a doubly linked list without using any existing implementation in Java.

- The double-linked list should be a generic data structure that can store elements of string data type.
- You must create a *Node* class that represents each element in the doubled-linked list.
- You must create a *LinkedList* class that represents a doubly linked list which should include methods for inserting, deleting, accessing specific elements, checking empty, returning size, and other operations you want to implement.
- The insertion operation should be done at the front of the list.
- The implementation should include error handling to handle errors such as deleting elements from an empty list and accessing out-of-bounds.

**5 marks**

You must give clear rationales and detailed explanations of your design and implementation in the report.

**5 marks**

### Task 2

You must design and implement a hash table based on a Java array (not any array list or existing implementation from the Java library) and achieve the collision solution by using the linear probing way.

- You must create a *LinearProbingHashTable* class that represents a hash table by using the linear probing way for collision resolution. The initial capacity of the array should not exceed 20.
- You must devise a hash function that can work well for string-type data. The hash function devised should minimize the occurrence of collisions. You must not use the Java built-in *hashCode* method, though you can experiment with it.
- The implementation can handle errors such as null keys or keys with unexpected formats.
- The implementation should include methods for inserting, searching, deleting, and accessing key-value pairs.
- The implementation of the inserting operation can resize the table efficiently according to the strategy you design if the hash table is too full.
- The implementation of the deleting operation can handle the situation when the key is not found.
- The implementation can keep track of the load factor of the hash table and display it after each insertion or deletion.
- The implementation of the searching operation can search for the key and return the corresponding value if the key is found.

**5 marks**

You must give clear rationales and detailed explanations of your design and implementation in the report.

**5 marks****Task 3**

You must design and implement a hash table based on the linked list and achieve the collision solution by using the separate chaining way.

- You must create a *ChainingHashTable* class that represents a hash table by using the separate chaining way for collision resolution.
- You must use the doubly linked list devised in task 1 to implement the separate chaining way. The capacity of the linked list of separate chaining should not exceed 8.
- You must devise a hash function that can work well for string-type data. The hashing strategy of the hash function should be designed differently from that of task 2 and should minimize the occurrence of collisions. You must not use the Java built-in *hashCode* method, though you can experiment with it.
- The implementation can handle errors such as null keys or keys with unexpected formats.
- The implementation should include methods for inserting, searching, deleting, and accessing key-value pairs, as well as determining load factor.
- The implementation of the inserting operation can resize the table efficiently if the hash table is too full.
- The implementation of the deleting operation can handle the situation when the key is not found.
- The implementation can keep track of the load factor of the hash table and display it after each insertion or deletion.
- The implementation of the searching operation can search for the key and return the corresponding value if the key is found.
- The implementation of the hash table can resize the table capacity according to the strategy you designed.

**5 marks**

You must give clear rationales and detailed explanations of your design and implementation in the report.

**5 marks****Task 4**

You must implement a main program that engages objects of both the *LinearProbingHashTable* class and the *ChainingHashTable* class.

- You must design a set of test cases to evaluate the functionality and correctness of two different hash tables.
  - Set the capacity of the hash table to a small value so that collisions are easy to occur.
  - Verify that each of the hash functions is working well.
  - Verify that each of the implemented methods is working correctly.
  - Verify that the implementations of the Linear Probing way and Separate Chaining way for collision solutions are working effectively.
- The inner structure of the generated hash tables should be clearly illustrated as the executed result of the program.

**4 marks**

You must give clear rationales and detailed explanations of your design and implementation in the report.

- Demonstrate the executed result of the program, including the generated hash table and corresponding test data.
- Contrast and analyze the two hash tables generated based on the same set of test cases given.
- Contrast and analyze the difference between the two hash functions you devised based on the same set of test cases given.
- Give a rationale and detailed analysis of the effects of two different strategies of collision solution.

**6 marks**  
**total 40 marks**

### Relevant quotation

*“There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult.”*

Professor Sir Tony Hoare

*1980 Turing Award Lecture; Communications of the ACM 24 (2), (February 1981): pp. 75-83*

Please try to do this the first way.

### Obtaining help

It is encouraged to request further clarification on what is required for this assignment. Please try to do this during normal contact time and avoid asking for such help in the last week before the deadline.

You can discuss the requirements and the material covered in the assignment with others but what you create must be all your own work. Be careful to avoid collusion.

Declare in your report any help you have received other than that from the module teaching team.

### Feedback

In addition to the written feedback that we aim to provide within the normal interval, you will be able to obtain fast, brief, verbal formative feedback and help on correcting your work at your practical classes.