CSC248 – Fundamentals of Data Structure
Academic Session Oct 2023 – Feb 2024
Lab Assignment – Review of OOP

Course Outcomes (CO)	LO1	LO2	LO3
CO1			
CO2	√	√	√
CO3			

1.1 Class `Land` has the following attributes and methods:

Attributes:

- id
- owner name
- house type
- area

i. Write the `Land` class and the following methods:

a) Default constructor.
b) Normal constructor that set all data with values given through the parameter.
c) Mutator/Setter method
d) Retriever method for each attribute.
e) Printer method using `toString()` defined method.
f) A processor method to calculate and return the tax amount. The tax of this type of land depends on its area, and the type of the house built on the land as shown in the following table:

House Type	Description	Tax rate (RM/m^3)
T	Terrace	10
S	Semi-Detached	15
B	Bungalow	20
C	Condominium	30

Details of land

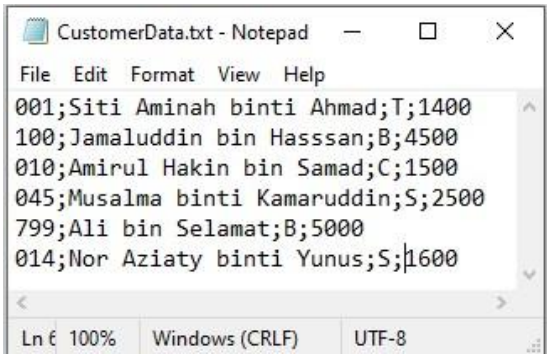
```java
public class Land {
    private String id;
    private String ownerName;
    private char houseType;
    private double area;

    public Land() {
        id = "";
        ownerName = "";
        houseType = ' ';
        area = 0.0;
    }

    public Land(String id, String ownerName, char houseType, double area) {
        this.id = id;
        this.ownerName = ownerName;
        this.houseType = houseType;
        this.area = area;
    }

    public String getId() {
        return this.id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getOwnerName() {
        return this.ownerName;
    }

    public void setOwnerName(String ownerName) {
        this.ownerName = ownerName;
    }

    public char getHouseType() {
        return this.houseType;
    }

    public void setHouseType(char houseType) {
        this.houseType = houseType;
    }
```

```
    public double getArea() {
        return this.area;
    }

    public void setArea(double area) {
        this.area = area;
    }

    public String toString() {
        return "ID: " + id + "\nOwner Name: " + ownerName + "\nHouse Type: " +
houseType + "\nArea: " + area;
    }

    public double calculateTax() {
        double tax = 0.0;

        if (houseType == 'T') {
            tax = 10 * area;
        } else if (houseType == 'S') {
            tax = 15 * area;
        } else if (houseType == 'B') {
            tax = 20 * area;
        } else if (houseType == 'C') {
            tax = 30 * area;
        }

        return tax;
    }
}
```

ii. Write an application program that performs the following:

a) Declare array of object for `Land` objects.

b) Given the input file named `customerData.txt` that includes the customers data such as id, owner name, house type and area. The following input text fil includes all record of customer for the `Land` class:



Write a program that reads each record from `customerData.txt` and store onto array of object `Land`.

c) Display a menu selection to select the following process:

```
------------------------------------------------------
                    Menu Selection


1. Sorting using Bubble Sort 2. Sorting using Insertion
Sort 3. Searching using Binary Search

Your Option: XX

------------------------------------------------------
```

****Details explanation:**

1-Sorting using Bubble Sort – Sort the list based on the tax price and display the list

2.Sorting using Insertion Sort – Sort the list based on id and display the list

3.Searching using Binary Search-Search the item from the list based on id and display the information detail.

```java
import java.io.BufferedReader;
import java.io.FileReader;
// import java.util.ArrayList;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner strInput = new Scanner(System.in);
        Scanner intInput = new Scanner(System.in);

        // ArrayList<Land> lands = new ArrayList<Land>();

        Land[] lands = null;

        // read customerData.txt
        try {
            BufferedReader br = new BufferedReader(new
FileReader("customerData.txt"));
            // get the number of lines in the file
            int count = 0;
            String line = br.readLine();
            while (line != null) {
                count++;
                line = br.readLine();
            }
            br.close();

            lands = new Land[count];
            count = 0;

            br = new BufferedReader(new FileReader("customerData.txt"));
            line = br.readLine();
            while (line != null) {
                String id = line.split(";")[0];
                String ownerName = line.split(";")[1];
                char houseType = line.split(";")[2].charAt(0);
                double area = Double.parseDouble(line.split(";")[3]);

                lands[count] = new Land(id, ownerName, houseType, area);
                count++;
                line = br.readLine();
            }

            br.close();
```

```
        } catch (Exception e) {
            System.out.println(e.getMessage() + "\nAt line: " +
e.getStackTrace()[0].getLineNumber());
        }

        System.out.println("---------------------------------------");
        System.out.println("Menu Selection");
        System.out.println("\n1. Sorting using Bubble Sort");
        System.out.println("2. Sorting using Insertion Sort");
        System.out.println("3. Searching using Binary Search");
        System.out.print("\nYour Option: ");
        int option = intInput.nextInt();

        System.out.println("---------------------------------------\n");

        // counter for the number of swaps
        int counter = 0;
        if (option == 1) {
            System.out.println("Sorting using Bubble Sort\n");

            // sort based on the tax price and display the list
            // O(n^2)
            // for (int i = 0; i < lands.size() - 1; i++) {
            // for (int j = 0; j < lands.size() - i - 1; j++) {
            // if (lands.get(j).calculateTax() > lands.get(j + 1).calculateTax())
{
            // Land temp = lands.get(j);
            // lands.set(j, lands.get(j + 1));
            // lands.set(j + 1, temp);
            // counter++;
            // }
            // }
            // }

            for (int i = 0; i < lands.length - 1; i++) {
                for (int j = 0; j < lands.length - i - 1; j++) {
                    if (lands[j].calculateTax() > lands[j + 1].calculateTax()) {
                        Land temp = lands[j];
                        lands[j] = lands[j + 1];
                        lands[j + 1] = temp;
                        counter++;
                    }
                }
            }
```

```
            for (int i = 0; i < lands.length; i++) {
                System.out.println(lands[i].toString());
                System.out.printf("Tax: RM %,.2f\n", lands[i].calculateTax());
                System.out.println();
            }

            // for (Land land : lands) {
            // System.out.println(land.toString());
            // System.out.printf("Tax: RM %,.2f\n", land.calculateTax());
            // System.out.println();
            // }

            System.out.println("This is sorted based on the tax price");

        } else if (option == 2) {
            System.out.println("Sorting using Insertion Sort\n");

            // sort based on id and display the list
            // for (int i = 1; i < lands.size(); i++) {
            // Land key = lands.get(i);
            // int j = i - 1;

            // while (j >= 0 && lands.get(j).getId().compareTo(key.getId()) > 0)
{
            // lands.set(j + 1, lands.get(j));
            // j--;
            // counter++;
            // }

            // lands.set(j + 1, key);

            // }

            // for (Land land : lands) {
            // System.out.println(land.toString());
            // System.out.printf("Tax: RM %,.2f\n", land.calculateTax());
            // System.out.println();
            // }

            for (int i = 1; i < lands.length; i++) {
                Land key = lands[i];
                int j = i - 1;
```

```
                while (j >= 0 && lands[j].getId().compareTo(key.getId()) > 0) {
                    lands[j + 1] = lands[j];
                    j--;
                    counter++;
                }

                lands[j + 1] = key;
            }

            for (int i = 0; i < lands.length; i++) {
                System.out.println(lands[i].toString());
                System.out.printf("Tax: RM %,.2f\n", lands[i].calculateTax());
                System.out.println();
            }

            System.out.println("This is sorted based on the ID");

        } else if (option == 3) {
            System.out.println("Searching using Binary Search\n");
            // sort based on id and display the list
            // for (int i = 1; i < lands.size(); i++) {
            // Land key = lands.get(i);
            // int j = i - 1;

            // while (j >= 0 && lands.get(j).getId().compareTo(key.getId()) > 0)
{
            // lands.set(j + 1, lands.get(j));
            // j--;
            // }

            // lands.set(j + 1, key);
            // }

            for (int i = 1; i < lands.length; i++) {
                Land key = lands[i];
                int j = i - 1;

                while (j >= 0 && lands[j].getId().compareTo(key.getId()) > 0) {
                    lands[j + 1] = lands[j];
                    j--;
                }

                lands[j + 1] = key;
            }
```

```
            System.out.print("Enter the ID to search: ");
            String id = strInput.nextLine();

            int low = 0;
            // int high = lands.size() - 1;
            int high = lands.length - 1;
            int mid = (low + high) / 2;

            // while (low <= high) {
            // if (lands.get(mid).getId().compareTo(id) < 0) {
            // low = mid + 1;
            // } else if (lands.get(mid).getId().compareTo(id) == 0) {
            // System.out.println(lands.get(mid).toString());
            // System.out.printf("Tax: RM %,.2f\n",
lands.get(mid).calculateTax());
            // break;
            // } else {
            // high = mid - 1;
            // }

            // mid = (low + high) / 2;
            // counter++;
            // }

            while (low <= high) {
                if (lands[mid].getId().compareTo(id) < 0) {
                    low = mid + 1;
                } else if (lands[mid].getId().compareTo(id) == 0) {
                    System.out.println();
                    System.out.println(lands[mid].toString());
                    System.out.printf("Tax: RM %,.2f\n",
lands[mid].calculateTax());
                    break;
                } else {
                    high = mid - 1;
                }

                mid = (low + high) / 2;
                counter++;
            }

            if (low > high) {
                System.out.println("ID not found!");
```

```
            } else {
                System.out.println("This is sorted based on the ID");
            }

        } else {
            System.out.println("Invalid option!");
        }

        if (option != 3)
            System.out.println("Number of swaps: " + counter);
        else
            System.out.println("\nNumber of try: " + counter);

        strInput.close();
        intInput.close();
    }
}
```

Sample Input/Output

```
---------------------------------------
Menu Selection

1. Sorting using Bubble Sort
2. Sorting using Insertion Sort
3. Searching using Binary Search

Your Option: 1
---------------------------------------

Sorting using Bubble Sort

ID: 001
Owner Name: Siti Aminah binti Ahmad
House Type: T
Area: 1400.0
Tax: RM 14,000.00

ID: 014
Owner Name: Nor Aziaty binti Yunus
House Type: S
Area: 1600.0
Tax: RM 24,000.00

ID: 045
Owner Name: Musalma binti Kamaruddin
House Type: S
Area: 2500.0
Tax: RM 37,500.00

ID: 010
Owner Name: Amirul Hakin bin Samad
House Type: C
Area: 1500.0
Tax: RM 45,000.00

ID: 100
Owner Name: Jamaluddin bin Hasssan
House Type: B
Area: 4500.0
Tax: RM 90,000.00
```

```
ID: 799
Owner Name: Ali bin Selamat
House Type: B
Area: 5000.0
Tax: RM 100,000.00

This is sorted based on the tax price
Number of swaps: 7
```

```
---------------------------------------
Menu Selection

1. Sorting using Bubble Sort
2. Sorting using Insertion Sort
3. Searching using Binary Search

Your Option: 2
---------------------------------------

Sorting using Insertion Sort

ID: 001
Owner Name: Siti Aminah binti Ahmad
House Type: T
Area: 1400.0
Tax: RM 14,000.00

ID: 010
Owner Name: Amirul Hakin bin Samad
House Type: C
Area: 1500.0
Tax: RM 45,000.00

ID: 014
Owner Name: Nor Aziaty binti Yunus
House Type: S
Area: 1600.0
Tax: RM 24,000.00

ID: 045
Owner Name: Musalma binti Kamaruddin
House Type: S
Area: 2500.0
Tax: RM 37,500.00

ID: 100
Owner Name: Jamaluddin bin Hasssan
House Type: B
Area: 4500.0
Tax: RM 90,000.00
```

```
ID: 799
Owner Name: Ali bin Selamat
House Type: B
Area: 5000.0
Tax: RM 100,000.00

This is sorted based on the ID
Number of swaps: 5
```

```
---------------------------------------
Menu Selection

1. Sorting using Bubble Sort
2. Sorting using Insertion Sort
3. Searching using Binary Search

Your Option: 3
---------------------------------------

Searching using Binary Search

Enter the ID to search: 799

ID: 799
Owner Name: Ali bin Selamat
House Type: B
Area: 5000.0
Tax: RM 100,000.00
This is sorted based on the ID

Number of try: 2
```