Practice Week 4

ArrayLists & Collections

Josh Hernandez

COMP 1020
Introductory Computer Science 2

University of Manitoba

Practice Problems

Hints and Expectations

- Scope: Week 4 (ArrayLists, Wrapper Classes, Collections basics).
- Assumes knowledge of Week 0-3 (Classes, Objects, File I/O).
- For “Predict the Output”, trace carefully on paper.
- For “Fill Code In”, strictly follow provided structure.
- For “Long Form”, practice writing complete classes/methods from scratch.

Predict Output - Question 1

Basic Operations

```
import java.util.ArrayList;

public class Q1 {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("A"); list = [A]
        list.add(0, "B"); list = [B]
        list.add("C");    list = [B, C]
        list.set(1, "D"); list = [B, D]
        list.remove(2);
        System.out.println(list);
    }
}
```

What is printed? output → [B, D]

Predict Output - Question 2

Remove Ambiguity

```
import java.util.ArrayList;

public class Q2 {
    public static void main(String[] args) {
        ArrayList<Integer> nums = new ArrayList<>();
        nums.add(5);
        nums.add(2);
        nums.add(5);
        nums.add(1); -> nums = [5,2,5,1]

        nums.remove(2); -> nums = [5,2,1] // Line A
        nums.remove(Integer.valueOf(5)); // Line B
                          ↳> [2,1]
        System.out.println(nums);
    }
}
        Output: [2,1]
```

What is printed? Note the difference between Line A and Line B.

Predict Output - Question 3

Autoboxing and math

```
import java.util.ArrayList;

public class Q3 {
    public static void main(String[] args) {
        ArrayList<Double> prices = new ArrayList<>();
        prices.add(10.5);
        prices.add(20.0);
        double total = prices.get(0) + prices.get(1);
        prices.add(total);

        System.out.println(prices.size() + " : " + prices.get(2));
    }
}
```

→ prices = [10.5 , 20.0]

= 30.5

→ prices = [10.5 , 20.0 , 30.5]

output → 3 : 30.5

What is printed?

Predict Output - Question 4

References and Mutation

```
import java.util.ArrayList;

class Box {
    int val;
    Box(int v) { val = v; }
    public String toString() { return "" + val; }
}

public class Q4 {
    public static void main(String[] args) {
        ArrayList<Box> list = new ArrayList<>();
        Box b = new Box(10);
        list.add(b);  list[0] = b (val = 10)
        list.add(b);  list[1] = b (val = 10)   } vì cả hai cùng trỏ vào 1 object nên khi val thay đổi thì bên kia cũng thay đổi theo.

        list.get(0).val = 99; → set val in b equals 99

        System.out.println(list); → output: [99, 99]
    }
}
```

Predict Output - Question 5

Loop Deletion Bug

```java
import java.util.ArrayList;

public class Q5 {
    public static void main(String[] args) {
        ArrayList<String> words = new ArrayList<>();
        words.add("bad");
        words.add("bad");
        words.add("good");

        // Attempt to remove all "bad"
        for (int i = 0; i < words.size(); i++) {
            if (words.get(i).equals("bad")) {
                words.remove(i);
            }
        }
        System.out.println(words);
    }
}
```

→ words = [bad, bad, good]

output: [bad, good]

i = 0

[bad, bad, good]

[bad, good]

khi bị remove ở index = 0 thì bad sẽ bị dồn lên và bad sẽ nằm ở index = 0 thay vì index = 1 như ban đầu

i = 1

[bad, good]

khi index = 1 bad sẽ bị skip và vòng lặp sẽ ghi nhận good và bỏ qua bad

What is printed? (Trace carefully!)

Predict Output - Question 6

IndexOf and Equals

```
import java.util.ArrayList;

class Item {
    String id;
    Item(String i) { id = i; }
    // No equals method!
}

public class Q6 {
    public static void main(String[] args) {
        ArrayList<Item> cart = new ArrayList<>();
        cart.add(new Item("A1"));
        int i = cart.indexOf(new Item("A1"));
        System.out.println(i);
    }
}
```

#1 Cart[0] = Item("A1");

#2 -1

indexOf(x) trong ArrayList sẽ duyệt list và so sánh bằng equals():

Nếu class không override equals(), thì nó dùng Object.equals() mặc định → so tham chiếu (reference), kiểu ==.

cart đang chứa new Item("A1") #1

Bạn đi tìm new Item("A1") #2 (object khác) → reference khác nhau → equals() mặc định trả false → không tìm thấy → -1

What is printed?

Predict Output - Question 7

Method Parameters

```
import java.util.ArrayList;

public class Q7 {
    public static void modify(ArrayList<Integer> data) {
        data.add(3); // lúc này data được trỏ tới nums và data = nums = [1]. Sau đó, data
        data = new ArrayList<>();   được add(3), vì vậy nums cũng được add(3) →
        data.add(4);                                                    data = nums = [1, 3]
    }   data lúc này được tạo như 1 list mới và không trỏ tới nums nữa. Mặc dù ta có
    public static void main(String[] args) {  data.add(4) và data = [4] thì ra khỏi method,
        ArrayList<Integer> nums = new ArrayList<>();   data sẽ biến mất.
        nums.add(1);  num = [1]
        modify(nums);
        System.out.println(nums);
    }
}
```

= nums

What is printed?

Predict Output - Question 8

Nested Structures

```
import java.util.ArrayList;

public class Q8 {
    public static void main(String[] args) {
        ArrayList<ArrayList<Integer>> matrix = new ArrayList<>();
        matrix.add(new ArrayList<>());
        matrix.get(0).add(5);
        matrix.get(0).add(10);

        System.out.println(matrix.get(0).size());
    }
}
```

output → 2

What is printed?

Predict Output - Question 9

String splitting and Lists

```
import java.util.ArrayList;

public class Q9 {
    public static void main(String[] args) {
        String s = "a,b,c";
        String[] parts = s.split(",");
        ArrayList<String> list = new ArrayList<>();
        for(String p : parts) list.add(p);

        list.set(1, list.get(2));
        System.out.println(list);
    }
}
```

parts = [a b c]

↳ list = [a, b, c]

→ list = [a, c, c]

↳ output: [a, c, c]

What is printed?

Predict Output - Question 10

Equality of Lists

```
import java.util.ArrayList;

public class Q10 {
    public static void main(String[] args) {
        ArrayList<String> a = new ArrayList<>();
        a.add("Hi");
        ArrayList<String> b = new ArrayList<>();
        b.add(new String("Hi"));

        // ArrayList.equals checks content equality
        System.out.println((a == b) + " " + a.equals(b));
    }
}
```

What is printed?

Fill Code In - Question 11

Array to ArrayList

Complete the method to copy all elements from an array to an ArrayList.

```
public static ArrayList<String> toList(String[] arr) {
    ArrayList<String> list = new ArrayList<>();

    // FILL IN HERE ----------------------
    for (int i = 0; i < arr.length; i++) {

        // Append current element
          list.add(arr[i]);
    }
    // ----------------------------------

    return list;
}
```

Fill Code In - Question 12

Find Maximum

Complete code to find the maximum value in a list of integers.

```
public static int findMax(ArrayList<Integer> nums) {
    if (nums.size() == 0) return 0; // Assume 0 if empty

    int max = nums.get(0);

    // FILL IN HERE ----------------------
        for( int i = 0; i < nums.size(); i++){
            if(nums.get(i) > max){
                max = nums.get(i);
            }
    // }-----------------------------------
    return max;
}
```

Fill Code In - Question 13

Manual Replace

Replace all occurrences of `"old"` with `"new"`.

```
public static void replaceAll(ArrayList<String> data,
                              String oldVal, String newVal) {
    // FILL IN HERE ----------------------
    for (int i = 0; i < data.size(); i++) {
        if (data.get(i).equals(oldVal)) {
            // Update at index i
            data.set(i, newVal);
        }
    }
    // ------------------------------------
}
```

Fill Code In - Question 14

Reverse List

Create a new list with elements in reverse order.

```
public static ArrayList<Integer> reverse(ArrayList<Integer> input) {
    ArrayList<Integer> result = new ArrayList<>();

    // FILL IN HERE ------------------------
    // Loop backwards for( int i = result.size() - 1 ; i >= 0 ; i--){
                          result. add ( input. get(i) );
                      }
    // ------------------------------------
    return result;
}
```

Fill Code In - Question 15

Safe Filter

Remove all negative numbers. **Note**: Be careful about index shifting!

```
public static void removeNegatives(ArrayList<Integer> nums) {
    // FILL IN HERE ----------------------
    int i = 0;
    while (i < nums.size()) {
        if (nums.get(i) < 0) {
            // Remove and DO NOT increment i
               nums.remove(i);
        } else {
            // Only increment if not removed
                i++;
        }
    }
    // ------------------------------------
}
```

Fill Code In - Question 16

Count Matches

Count how many strings start with the prefix string.

```
public static int countPrefix(ArrayList<String> list, String prefix) {
    int count = 0;
    // FILL IN HERE ----------------------
    for (String s : list) {
        // Check if s starts with prefix
           if ( s.length() >= prefix.length && s.substring(0, prefix.length).equals (prefix)){
                count ++;
           }
    }
    // -----------------------------------
    return count;
}
```

Fill Code In - Question 17

Merge Lists

Add all elements from `src` to the end of `dest`.

```
public static void appendList(ArrayList<String> dest,
                              ArrayList<String> src) {
    // FILL IN HERE ----------------------
       for (int i = 0 ; i < src.size() ; i++){
            dest.add( src.get(i));
       }
    // ----------------------------------
}
```

Fill Code In - Question 18

Sum of Evens

Calculate sum of even numbers in the list.

```
public static int sumEvens(ArrayList<Integer> nums) {
    int sum = 0;
    // FILL IN HERE ----------------------
    for (int n : nums) {
        // Check evenness
           if ( n % 2 == 0){
               sum += n;
    }       }
    // ------------------------------------
    return sum;
}
```

Fill Code In - Question 19

Deep Copy (Strings are immutable, so standard copy is fine)

Return a copy of the list.

```
public static ArrayList<String> copy(ArrayList<String> original) {
    // FILL IN HERE ----------------------

    // Create new list and add all from original
        ArrayList<String> copied = new ArrayList<>();
        if (int i= 0; i < original.size() ; i++){
    // ------------------------------------------
        copied.add(original.get(i));
}       }
        return copied;
```

Fill Code In - Question 20

Interleave

Combine two lists alternating elements: A1, B1, A2, B2...

```
public static ArrayList<String> zip(ArrayList<String> a,
                                    ArrayList<String> b) {
    ArrayList<String> result = new ArrayList<>();
    int size = Math.min(a.size(), b.size());
    // FILL IN HERE ----------------------
    for (int i = 0; i < size; i++) {
        result.add(a.get(i));
        result.add(b.get(i));
    }
    // ----------------------------------
    return result;
}
```

Long Form - Question 21

Statistics Class

Write a class `NumberStats` that stores numbers and computes stats.

- Field: `ArrayList<Double> numbers`
- Constructor: Initializes the list.
- Method: `void add(double d)`
- Method: `double getAverage()` (Return 0 if empty)

Long Form - Question 22

Unique Filter

Write a static method `getUnique` that takes an `ArrayList<String>` and returns a new `ArrayList<String>` containing only unique elements from the original list (removing duplicates).

Example: Input `["a", "b", "a"]` → Output `["a", "b"]`.

Long Form - Question 23

Grocery List

Write a class `GroceryList` with:

- Field: `ArrayList<String> items`
- Method: `void addItem(String item)` (Only add if not already in list)
- Method: `void remove(String item)`
- Method: `void printList()` (Print “1. Item” format)

Long Form - Question 24

Student Grade Filter

Assume a class `Student` exists with `getName()` and `getGrade()`.
Write a method `filterPassing`:

- Input: `ArrayList<Student>` and `double threshold`.
- Output: `ArrayList<Student>` (only those with grade $\geq$ threshold).

Long Form - Question 25

Matrix Row Sums

Write a method `rowSums` that takes a 2D structure `ArrayList<ArrayList<Integer>>` representing a matrix.
return a new `ArrayList<Integer>` containing the sum of each row.

Long Form - Question 26

Token Collector

Write a full program (with main) that:

- Reads words from standard input (Scanner) until "STOP" is entered.
- Stores them in an ArrayList.
- Prints them out in reverse order of entry, comma separated.

Long Form - Question 27

File Lines to List

Write a method `readLines` that takes a filename (String).

- Open the file.
- Read every line into an `ArrayList<String>`.
- Close file and return the list.
- Check exceptions (try/catch) - return empty list on error.

Long Form - Question 28

Min and Max Remover

Write a method `trimHelper` that takes a list of Doubles.

- Finds the minimum and maximum values.
- Removing ALL occurrences of min and max from the list.
- Return void (modify in place).

Long Form - Question 29

String Joiner

Write a method `join` that takes `ArrayList<String> list` and `String delimiter`. Return a single String combining all elements separated by the delimiter.
Example: Input `["a", "b"], "-"` → Output `"a-b"`.

Long Form - Question 30

Pairs to Sum

Write a method `findPairs` that takes a list of Integers and a target sum.
Print all pairs of indices (i, j) such that $i < j$ and $list.get(i) + list.get(j) == target$.