**COMP 203 Lab 6**

**List**

1. Implement List abstract data structure as a fixed size array in java. **(80pt)**

Implement Integer ArrayList class **(10pt)** and the following methods:

add(int i , int element): inserts “element” at the index of i. **(10pt)**

get(int i): returns the integer at the index of i. **(10pt)**

set(int i, int element): updates the element at the index of i with “element” and returns the old element value. **(10pt)**

remove(int i): removes the element at the index of i and returns the element that is removed. **(10pt)**

copyArray(int sizeofCurrentArray): copies the current array to the new array with the double size when the current array is full. This function returns the new array. **(10pt)**

isEmpty(): checks if the list is empty or not. **(5pt)**

isFull(): checks if the list is full or not. **(5pt)**

Test all these function in the main. Print the current condition of the list after every operation. **(10pt)**

**Submit ArrayList.java to Canvas.**

SOLUTION:

public class ArrayList { //**(5pt)**

private int[] array;

private int size;

private int length;

public ArrayList(int size) { //**(5pt)**

this.size = size;

this.array = new int[size];

this.length = 0;

}

**public void add(int index, int element) { //(10pt)**

if (index < 0 || index > length) {

throw new IndexOutOfBoundsException("Index out of bounds");

}

if (isFull()) {

array = copyArray( size);

}

for (int i = length; i > index; i--) {

array[i] = array[i - 1];

}

array[index] = element;

length++;

printList();

}

**public int get(int index) {**//**(10pt)**

if (index < 0 || index >= length) {

throw new IndexOutOfBoundsException("Index out of bounds");

}

return array[index];

}

**public int set(int index, int element) {**//**(10pt)**

if (index < 0 || index >= length) {

throw new IndexOutOfBoundsException("Index out of bounds");

}

int oldValue = array[index];

array[index] = element;

printList();

return oldValue;

}

**public int remove(int index) {**//**(10pt)**

if (index < 0 || index >= length) {

throw new IndexOutOfBoundsException("Index out of bounds");

}

int removedValue = array[index];

for (int i = index; i < length - 1; i++) {

array[i] = array[i + 1];

}

length--;

printList();

return removedValue;

}

**private int[] copyArray(int newSize) {** //**(10pt)**

int[] newArray = new int[2\*newSize];

System.arraycopy(array, 0, newArray, 0, length);

size = newSize;

return newArray;

}

public boolean isEmpty() { //**(5pt)**

return length == 0;

}

public boolean isFull() { //**(5pt)**

return length == size;

}

private void printList() {

System.out.print("Current List: ");

for (int i = 0; i < length; i++) {

System.out.print(array[i] + " ");

}

System.out.println();

}

**public static void main(String[] args) { //(10pt)**

ArrayList arrList = new ArrayList(5);

arrList.add(0, 1);

arrList.add(1, 2);

arrList.add(2, 3);

System.out.println("Element at index 1: " + arrList.get(1));

System.out.println("Old value at index 1: " + arrList.set(1, 4));

System.out.println("Removed element at index 0: " + arrList.remove(0));

System.out.println("Is the list empty? " + arrList.isEmpty());

System.out.println("Is the list full? " + arrList.isFull());

}

}

2. Use built-in List class of Java along with its methods add, remove, set, get in the main. Use string as a data type. You may create your list object from ArrayList class. Print the current condition of the list after every operation. (20pt)

**Submit Test.java to Canvas.**

**SOLUTION: add, remove, set, get: 5 pt for each one.**

import java.util.ArrayList;

import java.util.List;

public class Test {

public static void main(String[] args) {

List<String> stringList = new ArrayList<>();

// Adding elements to the list

stringList.add("Apple");

printList(stringList);

stringList.add("Banana");

printList(stringList);

stringList.add("Cherry");

printList(stringList);

// Getting element at index 1

String elementAtIndex1 = stringList.get(1);

System.out.println("Element at index 1: " + elementAtIndex1);

// Updating element at index 1

String oldValue = stringList.set(1, "Grapes");

System.out.println("Old value at index 1: " + oldValue);

printList(stringList);

// Removing element at index 0

String removedValue = stringList.remove(0);

System.out.println("Removed element at index 0: " + removedValue);

printList(stringList);

// Checking if the list is empty

System.out.println("Is the list empty? " + stringList.isEmpty());

}

private static void printList(List<String> list) {

System.out.print("Current List: ");

for (String element : list) {

System.out.print(element + " ");

}

System.out.println();

}

}