

Course: Programming Fundamental - ENSF 480

Lab #: Lab 5

Instructor: G. Gouri

Student Name: Daniel Rey

Lab Section: B01

Date Submitted: October 20, 2025

Exercise A

```
/*
 * File Name: BubbleSorter.java
 * Assignment: Lab 4, Exercise A
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

package exA;

import java.util.ArrayList;

public class BubbleSorter<E extends Number & Comparable<E>> implements Sorter<E>{
    public void sort(ArrayList<Item<E>> list){
        int s = list.size();
        for(int j=1; j<s; j++){
            for(int i=0; i<s-j; i++){
                if (list.get(i).getItem().compareTo(list.get(i+1).getItem())>0){
                    Item<E> temp = list.get(i);
                    list.set(i,list.get(i+1));
                    list.set(i+1,temp);
                }
            }
        }
    }
}

/*
 * File Name: InsertionSorter.java
 * Assignment: Lab 4, Exercise A
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

package exA;

import java.util.ArrayList;

public class InsertionSorter<E extends Number & Comparable<E>> implements Sorter<E>{
    public void sort(ArrayList<Item<E>> list){
        int s = list.size();
        for(int i=1; i<s; i++){
            Item<E> cur = list.get(i);
            for(int j=i; j>0 && list.get(j).getItem().compareTo(list.get(j-1).getItem())<0;j--){
                list.set(j,list.get(j-1));
                list.set(j-1,cur);
            }
        }
    }
}
```

```

/*
 * File Name: MyVector.java
 * Assignment: Lab 4, Exercise A
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

package exA;

import java.util.ArrayList;
import java.util.Iterator;

public class MyVector<E extends Number & Comparable<E>> {
    private ArrayList<Item<E>> storageM;
    private Sorter<E> sorter;

    MyVector(int n){
        storageM = new ArrayList<Item<E>>(n);
    }
    MyVector(ArrayList<Item<E>> arr){
        storageM = new ArrayList<Item<E>>(arr);
    }

    public void add(Item<E> value){
        storageM.add(value);
    }
    public void setSortStrategy(Sorter<E> s){
        sorter = s;
    }
    public void performSort(){
        sorter.sort(storageM);
    }
    public void display(){
        for(Iterator<Item<E>> itr = storageM.iterator(); itr.hasNext();){
            System.out.print(itr.next().getItem().toString()+" ");
        }
        System.out.println("");
    }
}

```

```

/*
 * File Name: Sorter.java
 * Assignment: Lab 4, Exercise A
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

package exA;

import java.util.ArrayList;

public interface Sorter<E extends Number & Comparable<E>> {
    abstract void sort(ArrayList<Item<E>> list);
}

```

```
}
```

Exercise B

```
/*
 * File Name: SelectionSorter.java
 * Assignment: Lab 4, Exercise B
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

package exA;

import java.util.ArrayList;

public class SelectionSorter<E extends Number & Comparable<E>> implements Sorter<E>{
    public void sort(ArrayList<Item<E>> list){
        int s = list.size();
        for(int j=0; j<s; j++){
            int min = j;
            for(int i=j+1; i<s; i++){
                if (list.get(i).getItem().compareTo(list.get(min).getItem())<0){
                    min = i;
                }
            }
            Item<E> temp = list.get(min);
            list.set(min,list.get(j));
            list.set(j,temp);
        }
    }
}
```

Exercise C

```
/*
 * File Name: DoubleArrayListSubject.java
 * Assignment: Lab 4, Exercise C
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

package exC;

import java.util.ArrayList;
import java.util.Iterator;

public class DoubleArrayListSubject implements Subject{
    private ArrayList<Observer> observers;
    private ArrayList<Double> data;

    DoubleArrayListSubject(){
        observers = new ArrayList<Observer>();
        data = new ArrayList<Double>();
    }

    public void addData(Double datum){
```

```

        data.add(datum);
        notifyAllObservers();
    }
    public void setData(Double datum, int i){
        data.set(i, datum);
        notifyAllObservers();
    }
    public void populate(ArrayList<Double> data){
        this.data = data;
        notifyAllObservers();
    }
    public void populate(double[] data){
        for (double x : data){
            this.data.add(x);
        }
        notifyAllObservers();
    }

    public void registerObserver(Observer o){
        observers.add(o);
        o.update(data);
    }
    public void remove(Observer o){
        observers.remove(o);
    }
    public void notifyAllObservers(){
        observers.forEach(o -> o.update(data));
    }

    public void display(){
        if (data.size()==0){
            System.out.println("Empty List: ...");
        } else {
            for(Iterator<Double> itr = data.iterator(); itr.hasNext();){
                System.out.print(itr.next().toString()+" ");
            }
            System.out.println("");
        }
    }
}

```

```

/*
 * File Name: FiveRowsTable_Observer.java
 * Assignment: Lab 4, Exercise C
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

```

```

package exC;

import java.util.ArrayList;

public class FiveRowsTable_Observer implements Observer{

```

```

private ArrayList<Double> state;

FiveRowsTable_Observer(Subject s){
    state = new ArrayList<Double>();
    s.registerObserver(this);
}

public void display(){
    int s = state.size();
    for(int i=0; i<5; i++){
        for(int j=0; i+5*j<s; j++){
            System.out.print(state.get(i+5*j).toString()+" ");
        }
        System.out.println("");
    }
    System.out.println("");
}

public void update(ArrayList<Double> state){
    this.state = state;
    System.out.println("\nNotification to Five-Rows Table Observer: Data Changed:");
    display();
}
}

```

```

/*
 * File Name: Observer.java
 * Assignment: Lab 4, Exercise C
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

```

```

package exC;

import java.util.ArrayList;

public interface Observer {
    abstract void update(ArrayList<Double> state);
}

```

```

/*
 * File Name: OneRow_Observer.java
 * Assignment: Lab 4, Exercise C
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

```

```

package exC;

import java.util.ArrayList;
import java.util.Iterator;

```

```

public class OneRow_Observer implements Observer{
    private ArrayList<Double> state;

    OneRow_Observer(Subject s){
        state = new ArrayList<Double>();
        s.registerObserver(this);
    }

    public void display(){
        for(Iterator<Double> itr = state.iterator(); itr.hasNext();){
            System.out.print(itr.next().toString()+" ");
        }
        System.out.println("");
    }
    public void update(ArrayList<Double> state){
        this.state = state;
        System.out.println("\nNotification to One-Row Observer: Data Changed:");
        display();
    }
}

```

```

/*
 * File Name: Subject.java
 * Assignment: Lab 4, Exercise C
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

```

```

package exC;

```

```

public interface Subject {
    abstract void registerObserver(Observer o);
    abstract void remove(Observer o);
    abstract void notifyAllObservers();
}

```

```

/*
 * File Name: ThreeColumnTable_Observer.java
 * Assignment: Lab 4, Exercise C
 * Lab Section: B01
 * Completed by: Daniel Rey
 * Submission Date: Oct 20, 2025
 */

```

```

package exC;

```

```

import java.util.ArrayList;
import java.util.Iterator;

```

```

public class ThreeColumnTable_Observer implements Observer{
    private ArrayList<Double> state;

```

```

ThreeColumnTable_Observer(Subject s){
    state = new ArrayList<Double>();
    s.registerObserver(this);
}

public void display(){
    for(Iterator<Double> itr = state.iterator(); itr.hasNext();){
        for(int i=0; i<2 && itr.hasNext(); i++){
            System.out.print(itr.next().toString()+" ");
        }
        System.out.println((itr.hasNext()) ? itr.next() : "");
    }
    System.out.println("");
}

public void update(ArrayList<Double> state){
    this.state = state;
    System.out.println("\nNotification to Three-Column Table Observer: Data Changed:");
    display();
}
}

```