Assignment 11

CS1083

Student Name: Omar Sebri
Student ID : 3722350

Code:

Part A:

```java
/**
 @author: omar sebri 3722350
Represents a student.
*/
public class Student implements Comparable<Student>{

    /**
    The last name of the student.
    */
    private String lastName;

    /**
    The first name of the student.
    */
    private String firstName;

    /**
    The student's ID number.
    */
    private int id;


    /**
    Constructs a student given their first and last name, and student ID.
    @param firstNameIn The first name of the student.
    @param lastNameIn The last name of the student.
    @param idIn The student's ID number.
    */
    public Student(String firstNameIn, String lastNameIn, int idIn){
        firstName = firstNameIn;
        lastName = lastNameIn;
        id = idIn;
    }

    /**
    Prints all the information about the student.
    @return The student's information.
    */
    public String toString(){
        return firstName + ", " + lastName + " (" + id + ")";
```

```
    }
    public String get_first_name(){
        return this.firstName;
    }
    public String get_last_name(){
        return this.lastName;
    }
    public int get_StudId(){
        return this.id;
    }

    @Override
    public int compareTo(Student s) {
        if (this.firstName.compareTo(s.get_first_name())>0)
            return 1;
        else if(this.firstName.compareTo(s.get_first_name())<0)
            return -1;
        else{
            if (this.lastName.compareTo(s.get_last_name())>0)
            return 1;
            else if(this.lastName.compareTo(s.get_last_name())<0)
            return -1;
            else{
                if (this.id>s.get_StudId())
            return 1;
                else if(this.id<s.get_StudId())
            return -1;
            }
        }
        return 0;
    }

}
```

Part B:

```
/** @author: omar sebri 3722350 */
public class ClassList{

    private StudentNode front,end;
    private int size;

    public ClassList(){
        this.front = null;
        this.end = null;
        this.size=0;
    }
```

```
    public void add(Student stud){
        StudentNode newNode = new StudentNode(stud);
        boolean inserted = false;
        if(this.front==null){
            this.front= newNode;
            this.end=newNode;
            size+=1;
        }
        else{
            StudentNode temp = this.front;

        while(newNode.data.compareTo(temp.data)>0 && temp.next!=null){
            temp=temp.next;
        }
        if(temp.next==null || (this.size==1 &&
newNode.data.compareTo(temp.data)<0 )){
            temp.next=newNode;
            //temp.prev.next=newNode;
            newNode.prev=temp;
            this.end=newNode;
            size+=1;
        }
        else{
            /*newNode.next=temp;
            newNode.prev=temp.prev;
            temp.prev=newNode;*/
            newNode.next=temp;
            newNode.prev=temp.prev;
            temp.prev.next=newNode;
            temp.prev=newNode;
            size+=1;
        }
    }
        /*else{
            StudentNode temp = this.front;
            while(temp.next!=null){
                if(newNode.data.compareTo(temp.data)<0){
                    newNode.prev=temp.prev;
                    newNode.next=temp;
                    temp.prev=newNode;
                    size+=1;
                    inserted=true;

                }
                temp=temp.next;
                }
            if(temp.next==null&&!inserted){
                if(newNode.data.compareTo(temp.data)<0){
                    newNode.prev=temp.prev;
```

```
                    newNode.next=temp;
                    temp.prev=newNode;
                    //this.end=temp;
                    size+=1;
                }
                else if(newNode.data.compareTo(temp.data)>0){
                    newNode.prev=temp;
                    temp.next=newNode;
                    this.end=newNode;
                    size+=1;
                }
            }

        }
        /*else if(this.size==1){
            if(stud.compareTo(this.front.data)>0){
                this.end.next=newNode;

            }
            else if(stud.compareTo(this.front.data)<0){
                this.end.data=this.front.data;
                this.front.data=stud.data;
                size+=1;
            }
        }*/
    }
    public void printer(){
        StudentNode temp = front;
        while(temp!=null){
            System.out.println(temp.data.toString());
            temp=temp.next;
        }
    }
    public Student[] getReveresedList(){
        Student[] array = new Student[this.size];
        StudentNode temp = end;
        int i=0;
        while(temp!=null){
            array[i]=temp.data;
            i++;
            temp=temp.prev;
        }
        return array;
    }
public void remove(Student studentOut){
     StudentNode temp = front;
     boolean done =true;
     while(temp!=null){
         if(temp.data.get_StudId()==studentOut.get_StudId()){
```

```
                if (temp.prev != null){
                    temp.prev.next = temp.next;
                    done = true; }
                else{
                    front = temp.next;
                    done=true; }
                if (temp.next != null){
                    temp.next.prev = temp.prev;
                    done=true; }
                else{
                    end = temp.prev;
                    done=true; }
            }
            temp=temp.next;
        }
        size-=1;
    }
    public int getNumStudent(){
        return this.size;
    }

    private class StudentNode{
        public Student data;
        public StudentNode next;
        public StudentNode prev;
        public StudentNode(Student dataIn){
            this.data= dataIn;
            this.next=null;
            this.prev=null;
        }
    }

}
```

Part C:

```
/** @author: omar sebri 3722350 */
public class Driver{
    public static void main(String[] args) {
        ClassList lista = new ClassList();
        Student s1 = new Student("ahmed","fayaz",21);
        Student s2 = new Student("john","dough",25);
        Student s3 = new Student("betty","white",26);
        Student s4 = new Student("harambe","ape",27);
        Student s5 = new Student("jazz","josh",35);

        lista.add(s1);
        lista.add(s4);
```

```
        lista.add(s2);
        lista.add(s3);
        lista.add(s5);
        System.out.println("printing the list:");
        lista.printer();
        System.out.println(" the number of students is:
"+lista.getNumStudent());


        Student [] array = lista.getReveresedList();
        System.out.println("List printed reversly:");
        for(int i =0; i<array.length;i++){
        System.out.println(array[i]);  }
        lista.remove(s4);
        System.out.println("printing the list:");
        lista.printer();
        System.out.println(" the number of students is:
"+lista.getNumStudent());


    }
}
```

Testing & outputs:

printing the list:

ahmed, fayaz (21)

betty, white (26)

harambe, ape (27)

john, dough (25)

jazz, josh (35)

the number of students is: 5

List printed reversly:

jazz, josh (35)

john, dough (25)

harambe, ape (27)

betty, white (26)

ahmed, fayaz (21)

printing the list:

ahmed, fayaz (21)

betty, white (26)

john, dough (25)

jazz, josh (35)

the number of students is: 4