NAME – AADITYA KHOT
PRN – 21610051
SUB JECT – JAVA
BATCH - S-5

Assignment- 8

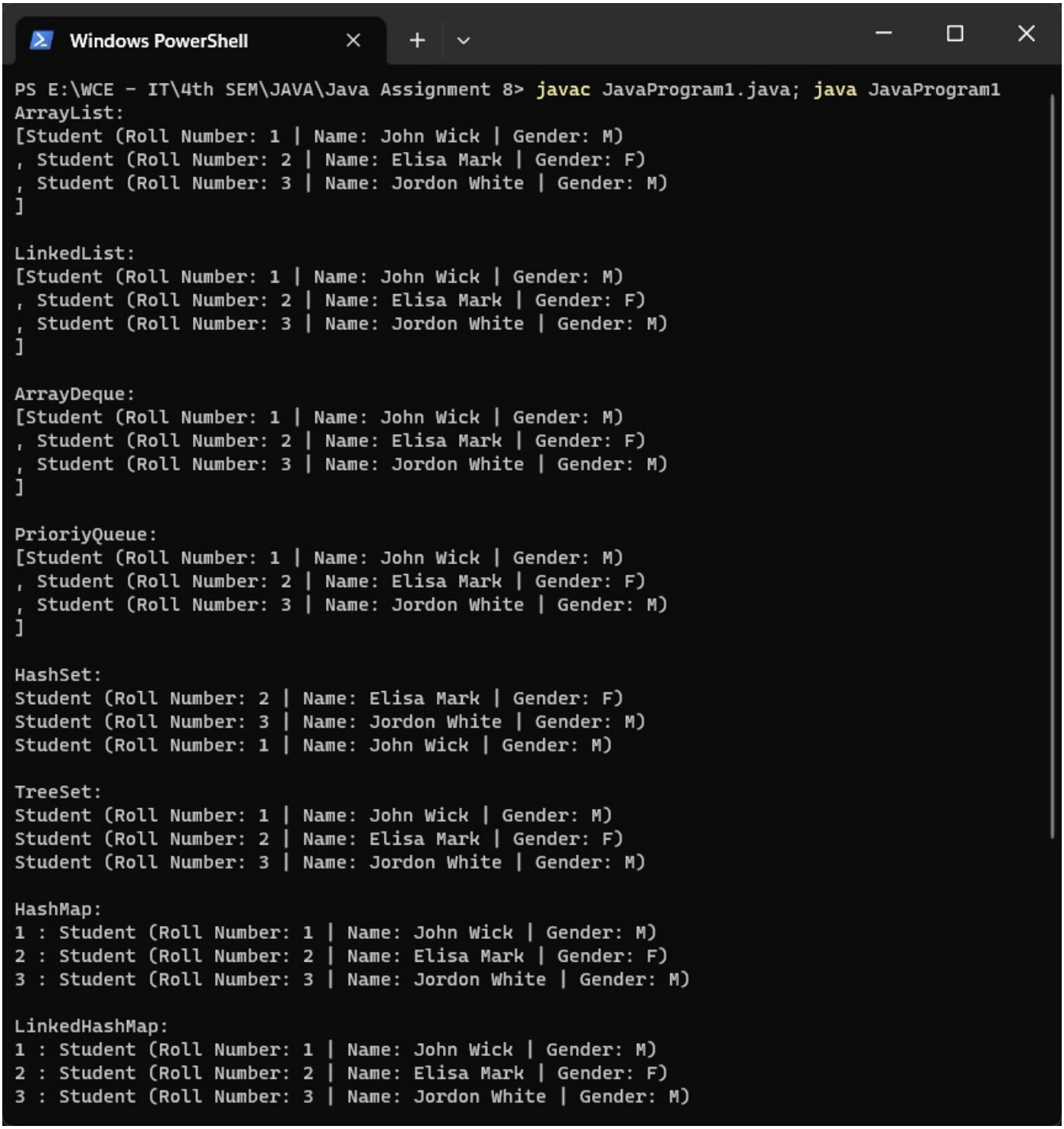
1. **Create objects of class student(roll number, name and gender), perform different operations on below collection components:**
    a. **ArrayList**
    b. **LinkedList**
    c. **ArrayDeque**
    d. **PriorityQueue**
    e. **HashSet**
    f. **TreeSet**
    g. **HashMap**
    h. **LinkedHashMap**

```
import java.sql.Time;
import java.util.*;

public class JavaProgram1 {
    public static void main(String[] args) {
        Student s1 = new Student(01, "John Wick", 'M');
        Student s2 = new Student(02, "Elisa Mark", 'F');
        Student s3 = new Student(03, "Jordon White", 'M');

        System.out.println("ArrayList: ");
        ArrayList<Student> arrayList = new ArrayList<Student>();
        arrayList.add(s1);
        arrayList.add(s2);
        arrayList.add(s3);
        System.out.println(arrayList + "\n");

        System.out.println("LinkedList: ");
        LinkedList<Student> linkedList = new LinkedList<Student>();
        linkedList.add(s1);
        linkedList.add(s2);
        linkedList.add(s3);
        System.out.println(linkedList + "\n");

        System.out.println("ArrayDeque: ");
        ArrayDeque<Student> arrayDeque = new ArrayDeque<Student>();
        arrayDeque.add(s1);
        arrayDeque.add(s2);
        arrayDeque.add(s3);
```

```
System.out.println(arrayDeque + "\n");
```

```
        System.out.println("PrioriyQueue: ");
        PriorityQueue<Student> priorityQueue = new
PriorityQueue<Student>();
        priorityQueue.add(s1);
        priorityQueue.add(s2);
        priorityQueue.add(s3);
        System.out.println(priorityQueue + "\n");

        System.out.println("HashSet: ");
        HashSet<Student> hashSet = new HashSet<Student>();
        hashSet.add(s1);
        hashSet.add(s2);
        hashSet.add(s3);

        Iterator hashSetIterator = hashSet.iterator();
        while(hashSetIterator.hasNext()) {
            System.out.print(hashSetIterator.next());
        }

        System.out.println("\nTreeSet: ");
        TreeSet<Student> treeSet = new TreeSet<Student>();
        treeSet.add(s1);
        treeSet.add(s2);
        treeSet.add(s3);

        Iterator<Student> treeSetIterator =treeSet.iterator();
        while(treeSetIterator.hasNext()) {
            System.out.print(treeSetIterator.next());
        }

        System.out.println("\nHashMap: ");
        HashMap<Integer, Student> hashMap = new HashMap<Integer,
Student>();
        hashMap.put(1, s1);
        hashMap.put(2, s2);
        hashMap.put(3, s3);
        for(HashMap.Entry    m    :    hashMap.entrySet())    {
            System.out.print(m.getKey() + " : " + m.getValue());
        }

        System.out.println("\nLinkedHashMap: ");
        LinkedHashMap<Integer, Student> linkedHashMap = new
LinkedHashMap<>();
        linkedHashMap.put(1,s1);
        linkedHashMap.put(2, s2);
        linkedHashMap.put(3, s3);
        for(Map.Entry e : linkedHashMap.entrySet()) {
            System.out.print(e.getKey() + " : " + e.getValue());
        }
    }
}

class Student implements Comparable<Student> {
    private int rNo;
    private String name;
```

```
    private char gender;

    Student(int rNo, String name, char gender) {
        this.rNo = rNo;
        this.name = name;
        this.gender = gender;
    }

    public int getrNo() {
        return rNo;
    }

    public String getName() {
        return name;
    }

    public char getGender() {
        return gender;
    }

    @Override
    public String toString() {
        return "Student (Roll Number: " + rNo + " | Name: " + name + "
| Gender: " + gender + ")\n";
    }

    @Override
    public int compareTo(Student o) {
        return (this.rNo - o.rNo);
    }
}
```



2. **Create object of class book (ISBN number, name and price), perform different operations on below collection components:**
    a. **ArrayList**
    b. **LinkedList**
    c. **ArrayDeque**
    d. **PriorityQueue**
    e. **HashSet**
    f. **TreeSet**
    g. **HashMap**
    h. **LinkedHashMap**

```
import java.sql.Time;
import java.util.*;

public class JavaProgram2 {
    public static void main(String[] args) {
        Book b1 = new Book(593202341, "All My Rage: A Nove", 1341.00);
        Book b2 = new Book(1684222273, "The Yellow Wallpaper", 442);
        Book b3 = new Book(9392099568L, "Sachin@50", 363.89);

        System.out.println("ArrayList: ");
        ArrayList<Book> arrayList = new ArrayList<Book>();
        arrayList.add(b1);
        arrayList.add(b2);
        arrayList.add(b3);
        System.out.println(arrayList + "\n");

        System.out.println("LinkedList: ");
        LinkedList<Book> linkedList = new LinkedList<Book>();
        linkedList.add(b1);
        linkedList.add(b2);
        linkedList.add(b3);
        System.out.println(linkedList + "\n");

        System.out.println("ArrayDeque: ");
        ArrayDeque<Book> arrayDeque = new ArrayDeque<Book>();
        arrayDeque.add(b1);
        arrayDeque.add(b2);
        arrayDeque.add(b3);
        System.out.println(arrayDeque + "\n");

        System.out.println("PrioriyQueue: ");
        PriorityQueue<Book> priorityQueue = new PriorityQueue<Book>();
        priorityQueue.add(b1);
        priorityQueue.add(b2);
        priorityQueue.add(b3);
        System.out.println(priorityQueue + "\n");

        System.out.println("HashSet: ");
        HashSet<Book> hashSet = new HashSet<Book>();
        hashSet.add(b1);
        hashSet.add(b2);
        hashSet.add(b3);

        Iterator hashSetIterator = hashSet.iterator();
        while(hashSetIterator.hasNext()) {
            System.out.print(hashSetIterator.next());
        }

        System.out.println("\nTreeSet: ");
        TreeSet<Book> treeSet = new TreeSet<Book>();
        treeSet.add(b1);
        treeSet.add(b2);
        treeSet.add(b3);

        Iterator<Book> treeSetIterator =treeSet.iterator();
        while(treeSetIterator.hasNext()) {
```

```
            System.out.print(treeSetIterator.next());
        }

        System.out.println("\nHashMap: ");
        HashMap<Integer, Book> hashMap = new HashMap<Integer, Book>();
        hashMap.put(1, b1);
        hashMap.put(2, b2);
        hashMap.put(3, b3);
        for(HashMap.Entry m : hashMap.entrySet()) {
            System.out.print(m.getKey() + " : " + m.getValue());
        }

        System.out.println("\nLinkedHashMap: ");
        LinkedHashMap<Integer, Book> linkedHashMap = new
LinkedHashMap<>();
        linkedHashMap.put(1,b1);
        linkedHashMap.put(2, b2);
        linkedHashMap.put(3, b3);
        for(Map.Entry e : linkedHashMap.entrySet()) {
            System.out.print(e.getKey() + " : " + e.getValue());
        }
    }
}

class Book implements Comparable<Book> {
    private long ISBN;
    private String name;
    private double price;

    Book(long isbn, String name, double price) {
        this.ISBN = isbn;
        this.name = name;
        this.price = price;
    }

    public long getISBN() {
        return ISBN;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    @Override
    public String toString() {
        return "Book (ISBN: " + ISBN + " | Name: " + name + " | Price:
" + price + ")\n";
    }

    @Override
    public int compareTo(Book o) {
```

```
        return this.ISBN == o.ISBN ? 1 : 0;
    }
}
```