Assignment- 08
22610011 - Anuja Suntnur

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

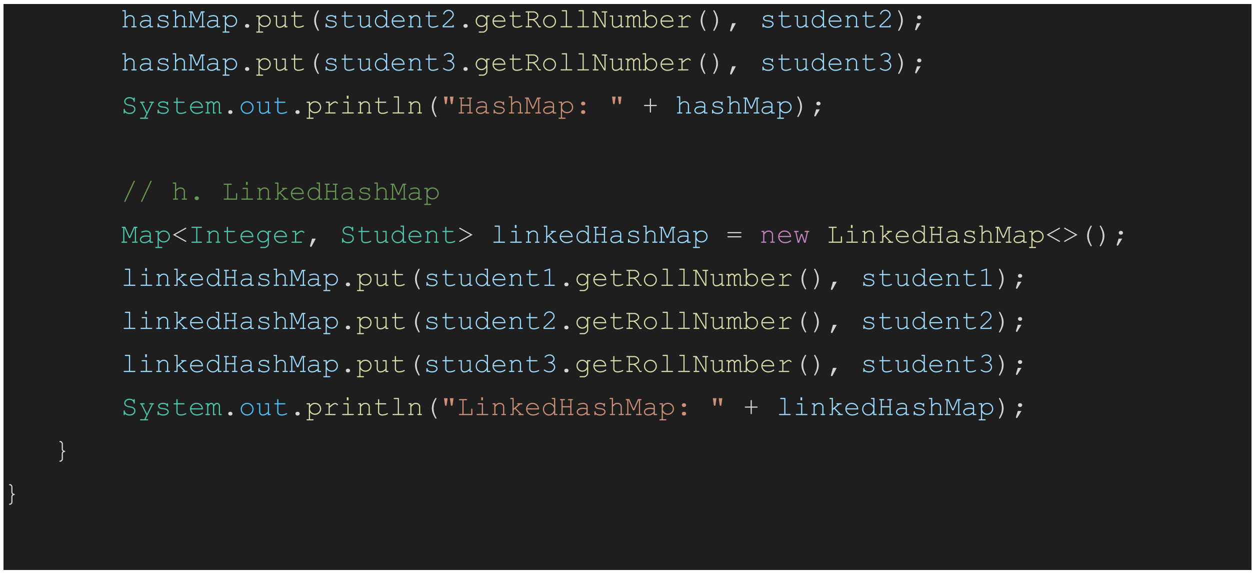
Ans:
1.Student.java file

```
import java.util.*;

public class Student {
   private int rollNumber;
   private String name;
   private String gender;

   public Student(int rollNumber, String name, String gender) {
       this.rollNumber = rollNumber;
       this.name = name;
       this.gender = gender;
   }

   public int getRollNumber() {
       return rollNumber;
   }
```

```
    public String getName() {
        return name;
    }

    public String getGender() {
        return gender;
    }

    @Override
    public String toString() {
        return "Student{" +
                "rollNumber=" + rollNumber +
                ", name='" + name + '\'' +
                ", gender='" + gender + '\'' +
                '}';
    }
}
```

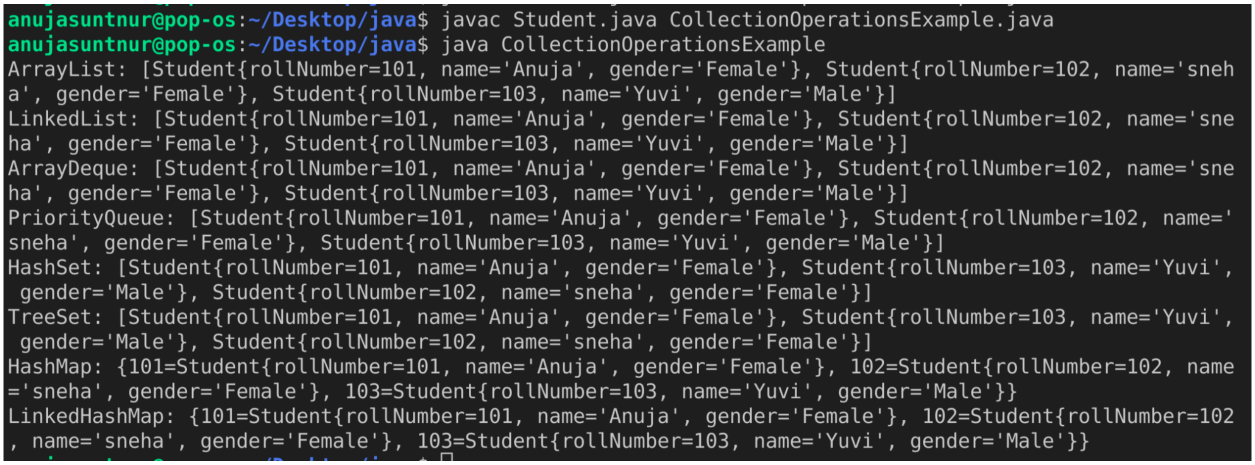
2. CollectionOperationsExample.java file

```
import java.util.*;

public class CollectionOperationsExample {
    public static void main(String[] args) {
        // Create objects of class Student
        Student student1 = new Student(101, "Anuja", "Female");
        Student student2 = new Student(102, "sneha", "Female");
        Student student3 = new Student(103, "Yuvi", "Male");

        // a. ArrayList
        List<Student> arrayList = new ArrayList<>();
        arrayList.add(student1);
        arrayList.add(student2);
        arrayList.add(student3);
        System.out.println("ArrayList: " + arrayList);
        // b. LinkedList
        List<Student> linkedList = new LinkedList<>();
        linkedList.add(student1);
        linkedList.add(student2);
        linkedList.add(student3);
        System.out.println("LinkedList: " + linkedList);

        // c. ArrayDeque
        Deque<Student> arrayDeque = new ArrayDeque<>();
        arrayDeque.add(student1);
        arrayDeque.add(student2);
        arrayDeque.add(student3);
        System.out.println("ArrayDeque: " + arrayDeque);

        // d. PriorityQueue
        Queue<Student> priorityQueue = new
PriorityQueue<>(Comparator.comparing(Student::getRollNumber));
        priorityQueue.add(student2);
        priorityQueue.add(student1);
        priorityQueue.add(student3);
        System.out.println("PriorityQueue: " + priorityQueue);

        // e. HashSet
        Set<Student> hashSet = new HashSet<>();
        hashSet.add(student1);
        hashSet.add(student2);
        hashSet.add(student3);
        System.out.println("HashSet: " + hashSet);

        // f. TreeSet
        Set<Student> treeSet = new
TreeSet<>(Comparator.comparing(Student::getName));
        treeSet.add(student2);
        treeSet.add(student1);
        treeSet.add(student3);
        System.out.println("TreeSet: " + treeSet);

        // g. HashMap
        Map<Integer, Student> hashMap = new HashMap<>();
        hashMap.put(student1.getRollNumber(), student1);
```



```
        hashMap.put(student2.getRollNumber(), student2);
        hashMap.put(student3.getRollNumber(), student3);
        System.out.println("HashMap: " + hashMap);

        // h. LinkedHashMap
        Map<Integer, Student> linkedHashMap = new LinkedHashMap<>();
        linkedHashMap.put(student1.getRollNumber(), student1);
        linkedHashMap.put(student2.getRollNumber(), student2);
        linkedHashMap.put(student3.getRollNumber(), student3);
        System.out.println("LinkedHashMap: " + linkedHashMap);
    }
}
```

Output :



```
anujasuntnur@pop-os:~/Desktop/java$ javac Student.java CollectionOperationsExample.java
anujasuntnur@pop-os:~/Desktop/java$ java CollectionOperationsExample
ArrayList: [Student{rollNumber=101, name='Anuja', gender='Female'}, Student{rollNumber=102, name='sneh
a', gender='Female'}, Student{rollNumber=103, name='Yuvi', gender='Male'}]
LinkedList: [Student{rollNumber=101, name='Anuja', gender='Female'}, Student{rollNumber=102, name='sne
ha', gender='Female'}, Student{rollNumber=103, name='Yuvi', gender='Male'}]
ArrayDeque: [Student{rollNumber=101, name='Anuja', gender='Female'}, Student{rollNumber=102, name='sne
ha', gender='Female'}, Student{rollNumber=103, name='Yuvi', gender='Male'}]
PriorityQueue: [Student{rollNumber=101, name='Anuja', gender='Female'}, Student{rollNumber=102, name='
sneha', gender='Female'}, Student{rollNumber=103, name='Yuvi', gender='Male'}]
HashSet: [Student{rollNumber=101, name='Anuja', gender='Female'}, Student{rollNumber=103, name='Yuvi',
 gender='Male'}, Student{rollNumber=102, name='sneha', gender='Female'}]
TreeSet: [Student{rollNumber=101, name='Anuja', gender='Female'}, Student{rollNumber=103, name='Yuvi',
 gender='Male'}, Student{rollNumber=102, name='sneha', gender='Female'}]
HashMap: {101=Student{rollNumber=101, name='Anuja', gender='Female'}, 102=Student{rollNumber=102, name
='sneha', gender='Female'}, 103=Student{rollNumber=103, name='Yuvi', gender='Male'}}
LinkedHashMap: {101=Student{rollNumber=101, name='Anuja', gender='Female'}, 102=Student{rollNumber=102
, name='sneha', gender='Female'}, 103=Student{rollNumber=103, name='Yuvi', gender='Male'}}
```

Q 2. Create objects 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

Ans:

1. Book.java file -

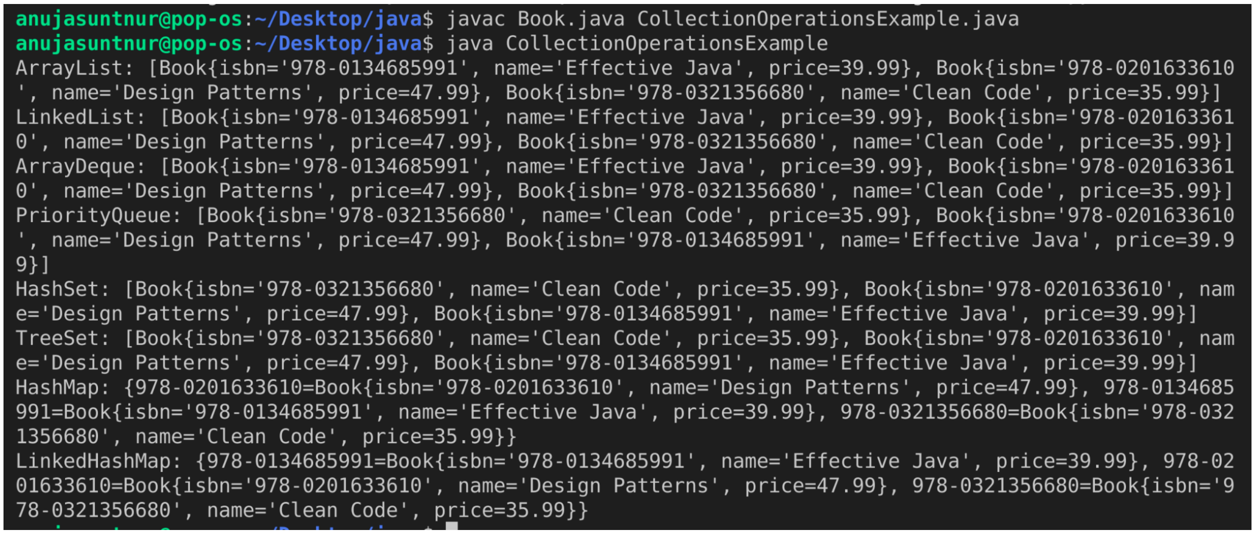
```java
import java.util.*;

class Book {
    private String isbn;
    private String name;
    private double price;

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

    public String 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 +
                '}';
    }
}
```

2. CollectionOperationsExample.java file -

```
import java.util.*;

public class CollectionOperationsExample {
   public static void main(String[] args) {
       // Create objects of class Book
       Book book1 = new Book("978-0134685991", "Effective Java", 39.99);
       Book book2 = new Book("978-0201633610", "Design Patterns", 47.99);
       Book book3 = new Book("978-0321356680", "Clean Code", 35.99);

       // a. ArrayList
       List<Book> arrayList = new ArrayList<>();
       arrayList.add(book1);
       arrayList.add(book2);
       arrayList.add(book3);
       System.out.println("ArrayList: " + arrayList);

       // b. LinkedList
       List<Book> linkedList = new LinkedList<>();
       linkedList.add(book1);
       linkedList.add(book2);
       linkedList.add(book3);
       System.out.println("LinkedList: " + linkedList);

       // c. ArrayDeque
       Deque<Book> arrayDeque = new ArrayDeque<>();
       arrayDeque.add(book1);
       arrayDeque.add(book2);
       arrayDeque.add(book3);
       System.out.println("ArrayDeque: " + arrayDeque);

       // d. PriorityQueue
       Queue<Book> priorityQueue = new
PriorityQueue<>(Comparator.comparing(Book::getPrice));
       priorityQueue.add(book2);
       priorityQueue.add(book1);
       priorityQueue.add(book3);
       System.out.println("PriorityQueue: " + priorityQueue);

       // e. HashSet
```

```
        Set<Book> hashSet = new HashSet<>();
        hashSet.add(book1);
        hashSet.add(book2);
        hashSet.add(book3);
        System.out.println("HashSet: " + hashSet);

        // f. TreeSet
        Set<Book> treeSet = new
TreeSet<>(Comparator.comparing(Book::getName));
        treeSet.add(book2);
        treeSet.add(book1);
        treeSet.add(book3);
        System.out.println("TreeSet: " + treeSet);

        // g. HashMap
        Map<String, Book> hashMap = new HashMap<>();
        hashMap.put(book1.getIsbn(), book1);
        hashMap.put(book2.getIsbn(), book2);
        hashMap.put(book3.getIsbn(), book3);
        System.out.println("HashMap: " + hashMap);

        // h. LinkedHashMap
        Map<String, Book> linkedHashMap = new LinkedHashMap<>();
        linkedHashMap.put(book1.getIsbn(), book1);
        linkedHashMap.put(book2.getIsbn(), book2);
        linkedHashMap.put(book3.getIsbn(), book3);
        System.out.println("LinkedHashMap: " + linkedHashMap);
    }
}
```

Output :



```
anujasuntnur@pop-os:~/Desktop/java$ javac Book.java CollectionOperationsExample.java
anujasuntnur@pop-os:~/Desktop/java$ java CollectionOperationsExample
ArrayList: [Book{isbn='978-0134685991', name='Effective Java', price=39.99}, Book{isbn='978-0201633610
', name='Design Patterns', price=47.99}, Book{isbn='978-0321356680', name='Clean Code', price=35.99}]
LinkedList: [Book{isbn='978-0134685991', name='Effective Java', price=39.99}, Book{isbn='978-020163361
0', name='Design Patterns', price=47.99}, Book{isbn='978-0321356680', name='Clean Code', price=35.99}]
ArrayDeque: [Book{isbn='978-0134685991', name='Effective Java', price=39.99}, Book{isbn='978-020163361
0', name='Design Patterns', price=47.99}, Book{isbn='978-0321356680', name='Clean Code', price=35.99}]
PriorityQueue: [Book{isbn='978-0321356680', name='Clean Code', price=35.99}, Book{isbn='978-0201633610
', name='Design Patterns', price=47.99}, Book{isbn='978-0134685991', name='Effective Java', price=39.9
9}]
HashSet: [Book{isbn='978-0321356680', name='Clean Code', price=35.99}, Book{isbn='978-0201633610', nam
e='Design Patterns', price=47.99}, Book{isbn='978-0134685991', name='Effective Java', price=39.99}]
TreeSet: [Book{isbn='978-0321356680', name='Clean Code', price=35.99}, Book{isbn='978-0201633610', nam
e='Design Patterns', price=47.99}, Book{isbn='978-0134685991', name='Effective Java', price=39.99}]
HashMap: {978-0201633610=Book{isbn='978-0201633610', name='Design Patterns', price=47.99}, 978-0134685
991=Book{isbn='978-0134685991', name='Effective Java', price=39.99}, 978-0321356680=Book{isbn='978-032
1356680', name='Clean Code', price=35.99}}
LinkedHashMap: {978-0134685991=Book{isbn='978-0134685991', name='Effective Java', price=39.99}, 978-02
01633610=Book{isbn='978-0201633610', name='Design Patterns', price=47.99}, 978-0321356680=Book{isbn='9
78-0321356680', name='Clean Code', price=35.99}}
```