Name – Mahemud Borgave

PRN – 23620005

Batch – S3

**Assignment No – 08**

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:

import java.util.\*;

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 "Roll Number: " + rollNumber + ", Name: " + name + ", Gender: " + gender;

    }

}

public class CollectionOperations {

    public static void main(String[] args) {

        // Create objects of Student class

        Student s1 = new Student(101, "John", "Male");

        Student s2 = new Student(102, "Alice", "Female");

        Student s3 = new Student(103, "Bob", "Male");

        Student s4 = new Student(104, "Emily", "Female");

        Student s5 = new Student(105, "David", "Male");

        // a. ArrayList

        ArrayList<Student> arrayList = new ArrayList<>();

        arrayList.add(s1);

        arrayList.add(s2);

        arrayList.add(s3);

        arrayList.add(s4);

        arrayList.add(s5);

        System.out.println("ArrayList:");

        displayStudents(arrayList);

        // b. LinkedList

        LinkedList<Student> linkedList = new LinkedList<>();

        linkedList.add(s1);

        linkedList.add(s2);

        linkedList.add(s3);

        linkedList.add(s4);

        linkedList.add(s5);

        System.out.println("\nLinkedList:");

        displayStudents(linkedList);

        // c. ArrayDeque

        ArrayDeque<Student> arrayDeque = new ArrayDeque<>();

        arrayDeque.add(s1);

        arrayDeque.add(s2);

        arrayDeque.add(s3);

        arrayDeque.add(s4);

        arrayDeque.add(s5);

        System.out.println("\nArrayDeque:");

        displayStudents(arrayDeque);

        // d. PriorityQueue

        PriorityQueue<Student> priorityQueue = new PriorityQueue<>(Comparator.comparing(Student::getRollNumber));

        priorityQueue.add(s3);

        priorityQueue.add(s2);

        priorityQueue.add(s5);

        priorityQueue.add(s4);

        priorityQueue.add(s1);

        System.out.println("\nPriorityQueue:");

        displayStudents(priorityQueue);

        // e. HashSet

        HashSet<Student> hashSet = new HashSet<>();

        hashSet.add(s1);

        hashSet.add(s2);

        hashSet.add(s3);

        hashSet.add(s4);

        hashSet.add(s5);

        System.out.println("\nHashSet:");

        displayStudents(hashSet);

        // f. TreeSet

        TreeSet<Student> treeSet = new TreeSet<>(Comparator.comparing(Student::getRollNumber));

        treeSet.add(s3);

        treeSet.add(s2);

        treeSet.add(s5);

        treeSet.add(s4);

        treeSet.add(s1);

        System.out.println("\nTreeSet:");

        displayStudents(treeSet);

        // g. HashMap

        HashMap<Integer, Student> hashMap = new HashMap<>();

        hashMap.put(s1.getRollNumber(), s1);

        hashMap.put(s2.getRollNumber(), s2);

        hashMap.put(s3.getRollNumber(), s3);

        hashMap.put(s4.getRollNumber(), s4);

        hashMap.put(s5.getRollNumber(), s5);

        System.out.println("\nHashMap:");

        displayStudents(hashMap.values());

        // h. LinkedHashMap

        LinkedHashMap<Integer, Student> linkedHashMap = new LinkedHashMap<>();

        linkedHashMap.put(s1.getRollNumber(), s1);

        linkedHashMap.put(s2.getRollNumber(), s2);

        linkedHashMap.put(s3.getRollNumber(), s3);

        linkedHashMap.put(s4.getRollNumber(), s4);

        linkedHashMap.put(s5.getRollNumber(), s5);

        System.out.println("\nLinkedHashMap:");

        displayStudents(linkedHashMap.values());

    }

    // Helper method to display students

    private static void displayStudents(Collection<Student> students) {

        for (Student student : students) {

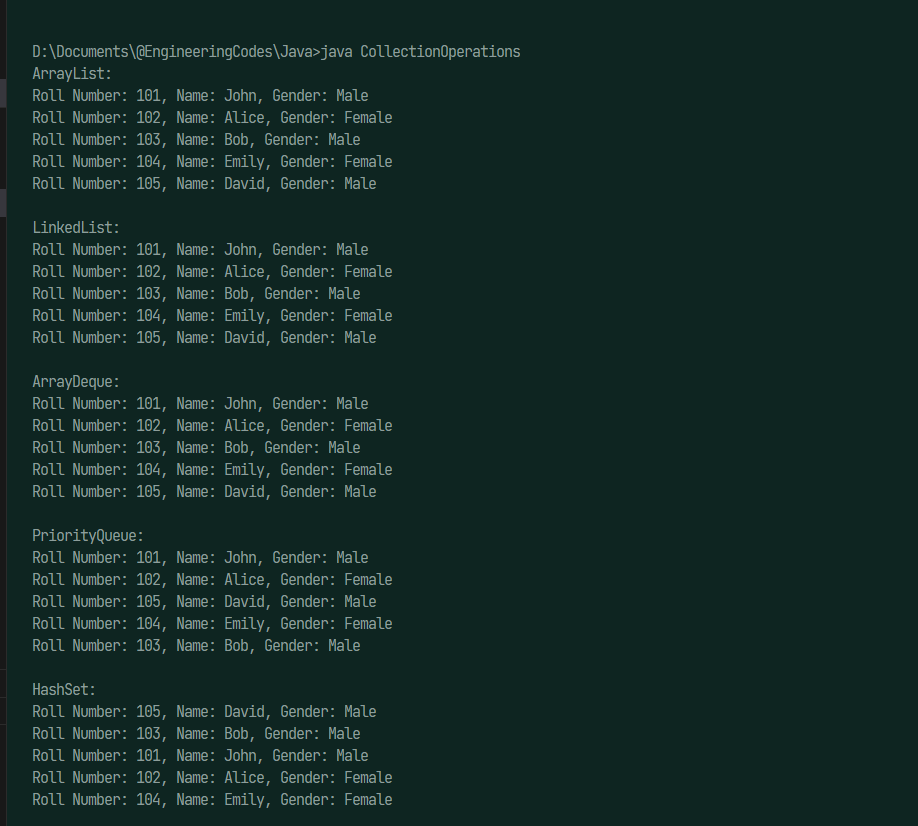
            System.out.println(student);

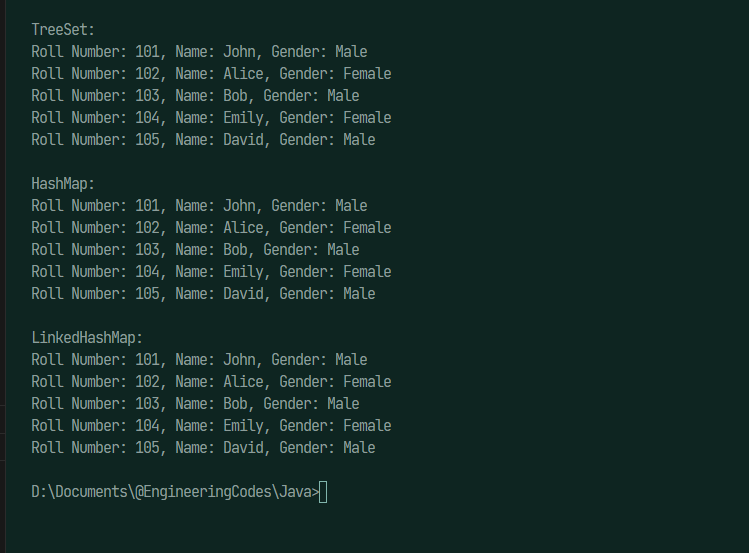
        }

    }

}

output –





1. Create object of class book (ISBN number, name and price), perform different operations on below collection components:

ArrayList

LinkedList

ArrayDeque

PriorityQueue

HashSet

TreeSet

HashMap

LinkedHashMap

Ans –

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 "ISBN: " + isbn + ", Name: " + name + ", Price: $" + price;

    }

}

public class CollectionOperations1 {

    public static void main(String[] args) {

        // Create objects of Book class

        Book b1 = new Book("978-1-56619-909-4", "The Great Gatsby", 12.99);

        Book b2 = new Book("978-1-56619-922-3", "To Kill a Mockingbird", 9.99);

        Book b3 = new Book("978-0-7475-5819-5", "Harry Potter and the Sorcerer's Stone", 17.99);

        Book b4 = new Book("978-0-618-15181-5", "Pride and Prejudice", 7.99);

        Book b5 = new Book("978-1-101-14501-2", "1984", 10.99);

        // a. ArrayList

        ArrayList<Book> arrayList = new ArrayList<>();

        arrayList.add(b1);

        arrayList.add(b2);

        arrayList.add(b3);

        arrayList.add(b4);

        arrayList.add(b5);

        System.out.println("ArrayList:");

        displayBooks(arrayList);

        // b. LinkedList

        LinkedList<Book> linkedList = new LinkedList<>();

        linkedList.add(b1);

        linkedList.add(b2);

        linkedList.add(b3);

        linkedList.add(b4);

        linkedList.add(b5);

        System.out.println("\nLinkedList:");

        displayBooks(linkedList);

        // c. ArrayDeque

        ArrayDeque<Book> arrayDeque = new ArrayDeque<>();

        arrayDeque.add(b1);

        arrayDeque.add(b2);

        arrayDeque.add(b3);

        arrayDeque.add(b4);

        arrayDeque.add(b5);

        System.out.println("\nArrayDeque:");

        displayBooks(arrayDeque);

        // d. PriorityQueue

        PriorityQueue<Book> priorityQueue = new PriorityQueue<>(Comparator.comparing(Book::getPrice));

        priorityQueue.add(b1);

        priorityQueue.add(b2);

        priorityQueue.add(b3);

        priorityQueue.add(b4);

        priorityQueue.add(b5);

        System.out.println("\nPriorityQueue:");

        displayBooks(priorityQueue);

        // e. HashSet

        HashSet<Book> hashSet = new HashSet<>();

        hashSet.add(b1);

        hashSet.add(b2);

        hashSet.add(b3);

        hashSet.add(b4);

        hashSet.add(b5);

        System.out.println("\nHashSet:");

        displayBooks(hashSet);

        // f. TreeSet

        TreeSet<Book> treeSet = new TreeSet<>(Comparator.comparing(Book::getPrice));

        treeSet.add(b1);

        treeSet.add(b2);

        treeSet.add(b3);

        treeSet.add(b4);

        treeSet.add(b5);

        System.out.println("\nTreeSet:");

        displayBooks(treeSet);

        // g. HashMap

        HashMap<String, Book> hashMap = new HashMap<>();

        hashMap.put(b1.getIsbn(), b1);

        hashMap.put(b2.getIsbn(), b2);

        hashMap.put(b3.getIsbn(), b3);

        hashMap.put(b4.getIsbn(), b4);

        hashMap.put(b5.getIsbn(), b5);

        System.out.println("\nHashMap:");

        displayBooks(hashMap.values());

        // h. LinkedHashMap

        LinkedHashMap<String, Book> linkedHashMap = new LinkedHashMap<>();

        linkedHashMap.put(b1.getIsbn(), b1);

        linkedHashMap.put(b2.getIsbn(), b2);

        linkedHashMap.put(b3.getIsbn(), b3);

        linkedHashMap.put(b4.getIsbn(), b4);

        linkedHashMap.put(b5.getIsbn(), b5);

        System.out.println("\nLinkedHashMap:");

        displayBooks(linkedHashMap.values());

    }

    // Helper method to display books

    private static void displayBooks(Collection<Book> books) {

        for (Book book : books) {

            System.out.println(book);

        }

    }

}

output –

