**CYCLE 4**

1. **Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.**

**Code:**

package\_graphics:

package package\_graphics;

interface interface\_package{

public float recArea(int l, int h);

public float cirArea(int r);

public float squArea(int a);

public float triArea(int l, int h);

}

public class package\_graphics implements interface\_package{

public float recArea(int l, int h){

return l\*h;

}

public float cirArea(int r){

return r\*r\*(float)3.14;

}

public float squArea(int a){

return a\*a;

}

public float triArea(int l, int h){

return l\*h\*(float)(.5);

}

}

Main\_graphics:

import package\_graphics.\*;

import java.util.\*;

public class main\_graphics{

public static void main(String[] args){

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 16/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

package\_graphics testObj = new package\_graphics();

int l,h,r,a,c,d;

Scanner s = new Scanner(System.in);

System.out.println("Enter the length for rectangle");

l = s.nextInt();

System.out.println("Enter the breadth for rectangle");

h = s.nextInt();

System.out.println("Enter the radius of circle");

r = s.nextInt();

System.out.println("Enter the side for Square");

a = s.nextInt();

System.out.println("Enter the breadth for triangle");

c = s.nextInt();

System.out.println("Enter the height for triangle");

d = s.nextInt();

System.out.println("Area of reactangle: "+testObj.recArea(l,h));

System.out.println("Area of circle: "+testObj.cirArea(r));

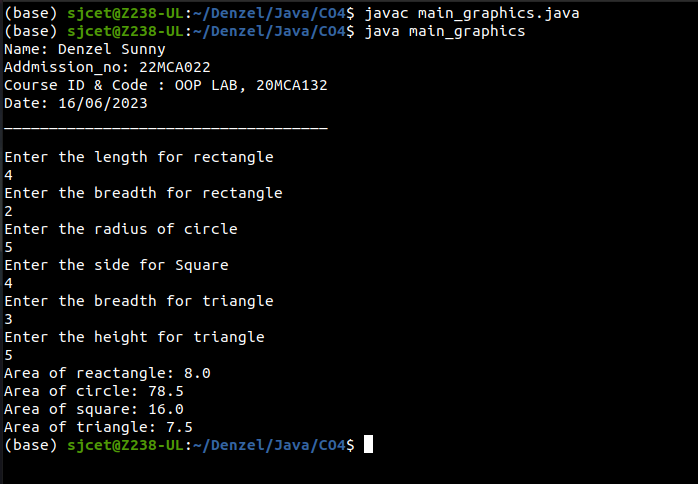
System.out.println("Area of square: "+testObj.squArea(a));

System.out.println("Area of triangle: "+testObj.triArea(c,d));

}

}

**Output:**



1. **Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers**

**Code:**

Arithmetic:

Addition:

package arithmetic;

public class Addition implements Arithmetic {

public double calculate(double a, double b) {

return a + b;

}

}

Arithmetic:

package arithmetic;

public interface Arithmetic {

double calculate(double a, double b);

}

Division:

package arithmetic;

public class Division implements Arithmetic {

public double calculate(double a, double b) {

if (b != 0) {

return a / b;

} else {

throw new ArithmeticException("Cannot divide by zero");

}

}

}

Multiplication:

package arithmetic;

public class Multiplication implements Arithmetic {

public double calculate(double a, double b) {

return a \* b;

}

}

Subtraction:

package arithmetic;

public class Subtraction implements Arithmetic {

public double calculate(double a, double b) {

return a - b;

}

}

Main:

import arithmetic.\*;

public class Q2 {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

double num1 = 10;

double num2 = 5;

Arithmetic addition = new Addition();

double sum = addition.calculate(num1, num2);

System.out.println("Sum: " + sum);

Arithmetic subtraction = new Subtraction();

double difference = subtraction.calculate(num1, num2);

System.out.println("Difference: " + difference);

Arithmetic multiplication = new Multiplication();

double product = multiplication.calculate(num1, num2);

System.out.println("Product: " + product);

Arithmetic division = new Division();

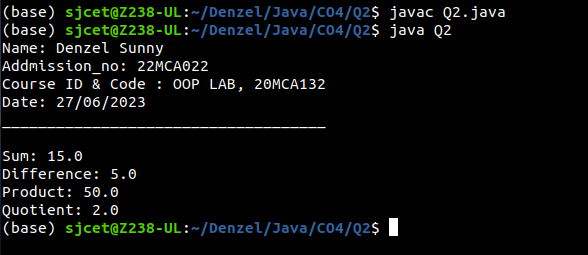
double quotient = division.calculate(num1, num2);

System.out.println("Quotient: " + quotient);

}

}

**Output:**



1. **Write a user defined exception class to authenticate the user name and password.**

**Code:**

import java.util.Scanner;

class authException extends Exception{

public authException(String s){

super(s);

}

}

public class Userauthentication{

public static void main(String[] args){

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 21/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

String username = "student";

String passcode = "student123";

String user\_name,password;

Scanner sc = new Scanner(System.in);

try

{

System.out.println("Enter username: ");

user\_name = sc.nextLine();

System.out.println("Enter password: ");

password = sc.nextLine();

if(username.equals(user\_name) && passcode.equals(password)){

System.out.println("Authentication successful...");

}

else

throw new authException("Invalid user credentials");

}

catch(authException e)

{

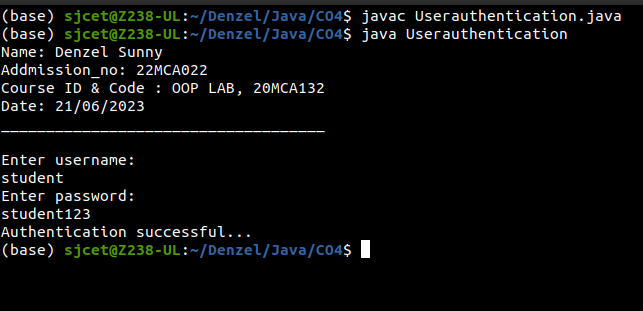
System.out.println("Exception caught "+e);

}

}

}

**Output:**



1. **Find the average of N positive integers, raising a user defined exception for each negative input**.

**Code:**

import java.util.Scanner;

class NegException extends Exception{

public NegException(String s){

super(s);

}

}

public class average {

public static void main(String[] args){

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 21/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

int i;

double sum=0,avg=0;

Scanner sc=new Scanner(System.in);

System.out.println("Enter n numbers:");

int n=sc.nextInt();

for(i=1;i<=n;i++){

try{

System.out.println("Enter number"+i);

int a=sc.nextInt();

if(a<0){

i--;

throw new NegException("Negative numbers not allowed,Try again");

}

else{

sum=sum+a;

}

}

catch(NegException e){

System.out.println("NEGETIVE EXCEPTION OCCURED:"+e);

}

}

avg=sum/n;

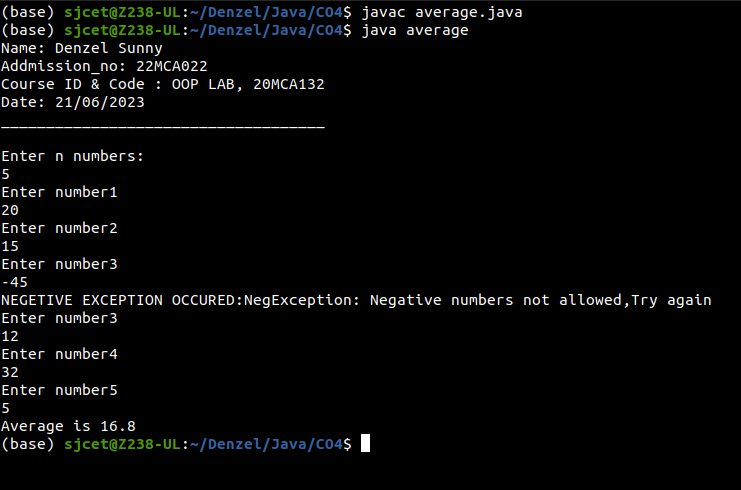
System.out.println("Average is "+avg);

sc.close();

}

}

**Output:**



1. **Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)**

**Code:**

class MultiplicationTable implements Runnable {

@Override

public void run() {

System.out.println("Multiplication Table of 5:");

for (int i = 1; i <= 10; i++) {

System.out.println("5 \* " + i + " = " + (5 \* i));

}

}

}

class PrimeNumbers implements Runnable {

@Override

public void run() {

System.out.println("First 10 Prime Numbers:");

int count = 0;

int num = 2;

while (count < 10) {

if (isPrime(num)) {

System.out.print(num + " ");

count++;

}

num++;

}

System.out.println();

}

private boolean isPrime(int num) {

if (num < 2) {

return false;

}

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

}

public class ThreadExample1 {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 23/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

MultiplicationTable multiplicationTable = new MultiplicationTable();

PrimeNumbers primeNumbers = new PrimeNumbers();

Thread thread1 = new Thread(multiplicationTable);

Thread thread2 = new Thread(primeNumbers);

thread1.start();

try {

thread1.join();

} catch (InterruptedException e) {

e.printStackTrace();

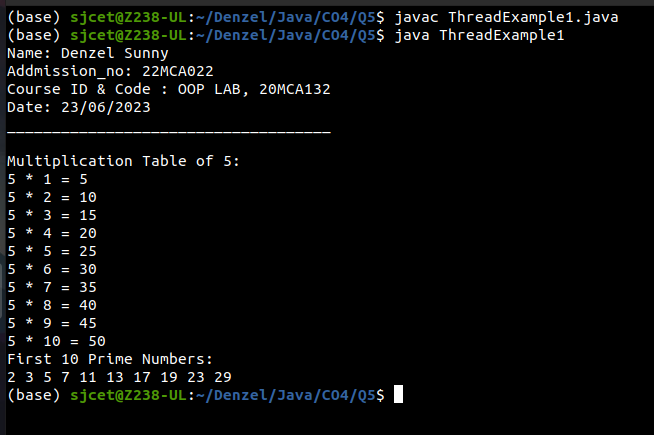
}

thread2.start();

}

}

**Output:**



1. **Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)**

**Code:**

import java.util.Scanner;

class Fibonacci implements Runnable {

private int count;

public Fibonacci(int count) {

this.count = count;

}

@Override

public void run() {

System.out.println("Fibonacci numbers:");

int num1 = 0;

int num2 = 1;

System.out.println(num1);

System.out.println(num2);

for (int i = 2; i < count; i++) {

int fib = num1 + num2;

System.out.println(fib);

num1 = num2;

num2 = fib;

}

}

}

class EvenNumber implements Runnable {

private int start;

private int end;

public EvenNumber(int start, int end) {

this.start = start;

this.end = end;

}

@Override

public void run() {

System.out.println("Even numbers from " + start + " to " + end + ":");

for (int i = start; i <= end; i++) {

if (i % 2 == 0) {

System.out.println(i);

}

}

}

}

public class multiThread {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 21/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the count of Fibonacci numbers: ");

int fibonacciCount = scanner.nextInt();

System.out.print("Enter the start of the range for even numbers: ");

int start = scanner.nextInt();

System.out.print("Enter the end of the range for even numbers: ");

int end = scanner.nextInt();

scanner.close();

Thread fibonacciThread = new Thread(new Fibonacci(fibonacciCount));

Thread evenNumberThread = new Thread(new EvenNumber(start, end));

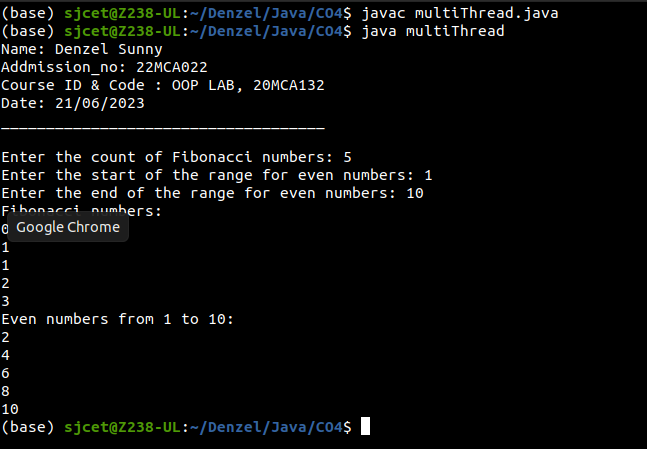
fibonacciThread.start();

evenNumberThread.start();

}

}

**Output:**



1. **Producer/Consumer using ITC**

**Code:**

import java.util.LinkedList;

class Producer implements Runnable{

LinkedList<Integer> list;

Producer(LinkedList<Integer> list){

this.list = list;

}

@Override

public void run() {

for(int i = 1; i <=7; i++){

synchronized(list) {

while(list.size() >= 1){

System.out.println("Waiting as queue is full..");

try {

list.wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("Adding to queue- " + Thread.currentThread().getName() + " " + i);

list.add(i);

list.notify();

}

}

}

}

class Consumer implements Runnable{

LinkedList<Integer> list;

Consumer(LinkedList<Integer> list){

this.list = list;

}

@Override

public void run() {

for(int i = 1; i <= 7; i++){

synchronized(list) {

while(list.size() < 1){

System.out.println("Waiting as queue is empty..");

try {

list.wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("Consuming from queue- " + Thread.currentThread().getName() + " " + list.remove());

list.notify();

}

}

}

}

public class ProducerConsumerITC {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 23/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

LinkedList<Integer> list = new LinkedList<Integer>();

Thread t1 = new Thread(new Producer(list), "Producer");

Thread t2 = new Thread(new Consumer(list), "Consumer");

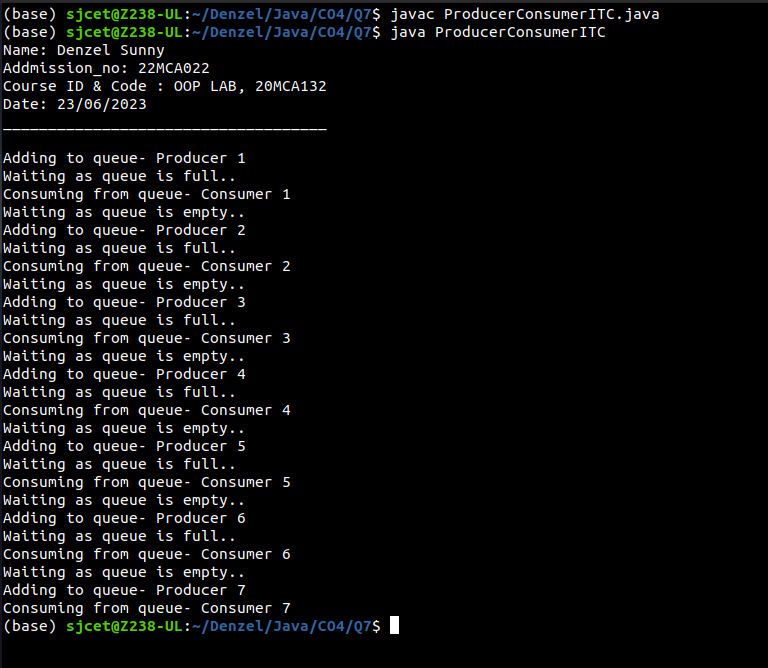
t1.start();

t2.start();

}

}

**Output:**



1. **Program to create a generic stack and do the Push and Pop operations.**

**Code:**

import java.util.Scanner;

public class StackExample {

int top=-1,ch,item,i;

int a[] = new int[10];

Scanner sc = new Scanner(System.in);

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 23/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

StackExample obj = new StackExample ();

obj.stack();

}

public void stack(){

System.out.println("Enter the size of the array:");

int N=sc.nextInt();

System.out.println("\t CHOICES : ");

System.out.println("\n 1.push \n 2.pop \n 3.exit \n");

while(ch<3) {

System.out.println("\n Enter your choice:");

ch=sc.nextInt();

switch(ch){

case 1:

System.out.println("Enter the element to be inserted:");

item=sc.nextInt();

if(top==N-1) {

System.out.println("stack overflow!");

}

else {

top++;

a[top]=item;

}

break;

case 2:

if(top==-1) {

System.out.println("stack is empty");

}

else {

item=a[top];

top--;

System.out.println("deleted element is:" +item);

}

break;

case 3 : break;

default : System.out.println("\n Invalid choice");

}

if(top < 0){

System.out.println("\n stack is empty");

}

else{

System.out.println("\n stack is \n");

for(i=top;i>=0;i--){

System.out.println(a[i]);

}

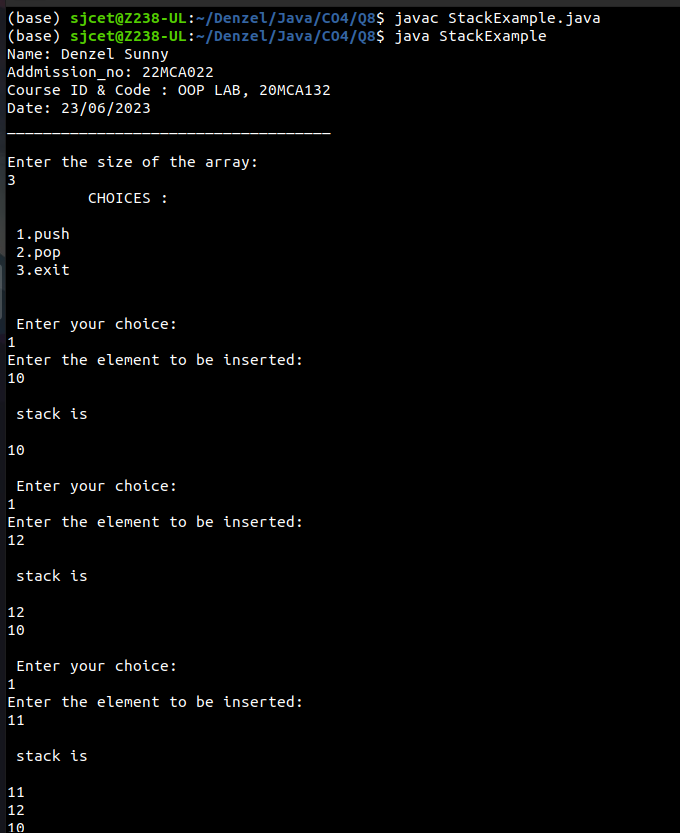
}

}

}

}

**Output:**



1. **Using generic method perform Bubble sort.**

**Code:**

import java.util.Arrays;

class BubbleSort {

public static <T extends Comparable<T>> void bubbleSort(T[] array) {

int n = array.length;

boolean swapped;

for (int i = 0; i < n - 1; i++) {

swapped = false;

for (int j = 0; j < n - i - 1; j++) {

if (array[j].compareTo(array[j + 1]) > 0) {

T temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

swapped = true;

}

}

if (!swapped) {

break;

}

}

}

}

public class BubbleSortExample {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 23/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Integer[] intArray = {5, 2, 8, 1, 9};

System.out.println("Before sorting: " + Arrays.toString(intArray));

BubbleSort.bubbleSort(intArray);

System.out.println("After sorting: " + Arrays.toString(intArray));

String[] stringArray = {"apple", "banana", "pear", "orange"};

System.out.println("Before sorting: " + Arrays.toString(stringArray));

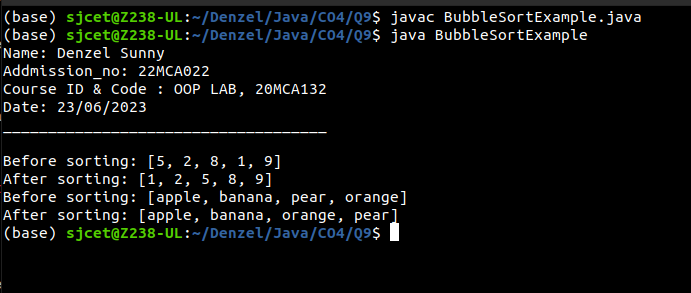
BubbleSort.bubbleSort(stringArray);

System.out.println("After sorting: " + Arrays.toString(stringArray));

}

}

**Output:**



1. **Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.**

**Code:**

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

public class ArrayListExample {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 23/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

List<String> stringList = new ArrayList<>();

stringList.add("Apple");

stringList.add("Banana");

stringList.add("Orange");

stringList.add("Mango");

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

System.out.println("Size: " + stringList.size());

System.out.println("Is Empty? " + stringList.isEmpty());

String element = stringList.get(2);

System.out.println("Element at index 2: " + element);

boolean contains = stringList.contains("Banana");

System.out.println("Contains 'Banana'? " + contains);

int index = stringList.indexOf("Mango");

System.out.println("Index of 'Mango': " + index);

Collections.sort(stringList);

System.out.println("Sorted ArrayList: " + stringList);

stringList.remove("Orange");

System.out.println("After removing 'Orange': " + stringList);

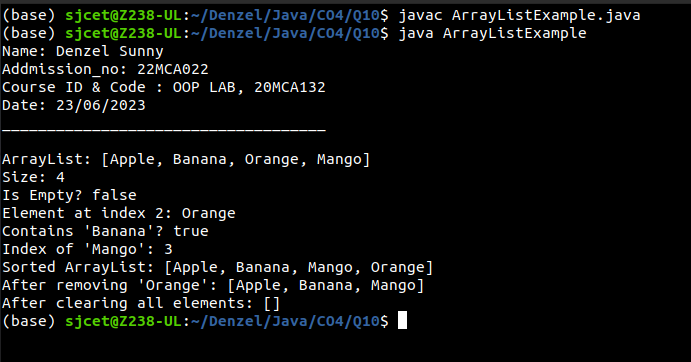
stringList.clear();

System.out.println("After clearing all elements: " + stringList);

}

}

**Output:**



1. **Program to remove all the elements from a linked list**

**Code:**

import java.util.LinkedList;

public class LinkedListClearExample {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

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

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

linkedList.add("Grapes");

System.out.println("Before clearing: " + linkedList);

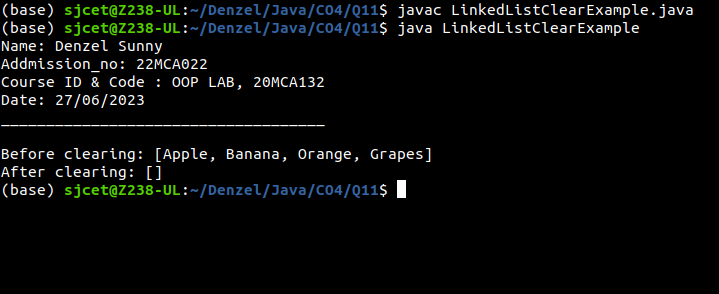
linkedList.clear();

System.out.println("After clearing: " + linkedList);

}

}

**Output:**



1. **Program to remove an object from the Stack when the position is passed as parameter**

**Code:**

import java.util.Stack;

public class StackRemove {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Stack<String> stack = new Stack<>();

stack.push("Apple");

stack.push("Banana");

stack.push("Orange");

stack.push("Grapes");

System.out.println("Before removing: " + stack);

int positionToRemove = 2; // Position starts from 1

removeByPosition(stack, positionToRemove);

System.out.println("After removing: " + stack);

}

public static void removeByPosition(Stack<String> stack, int position) {

if (position < 1 || position > stack.size()) {

System.out.println("Invalid position");

return;

}

Stack<String> tempStack = new Stack<>();

for (int i = 1; i < position; i++) {

tempStack.push(stack.pop());

}

stack.pop();

while (!tempStack.isEmpty()) {

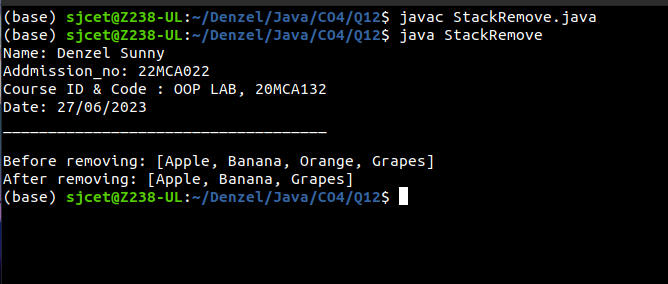
stack.push(tempStack.pop());

}

}

}

**Output:**



1. **Program to demonstrate the creation of queue object using the PriorityQueue class**

**Code:**

import java.util.PriorityQueue;

import java.util.Queue;

import java.util.Scanner;

public class PriorityQueueExample {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Queue<Integer> queue = new PriorityQueue<>();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements: ");

int numElements = scanner.nextInt();

System.out.println("Enter the elements:");

for (int i = 0; i < numElements; i++) {

int element = scanner.nextInt();

queue.offer(element);

}

System.out.println("Queue elements: " + queue);

System.out.println("Processing elements:");

while (!queue.isEmpty()) {

int element = queue.poll();

System.out.println("Processing element: " + element);

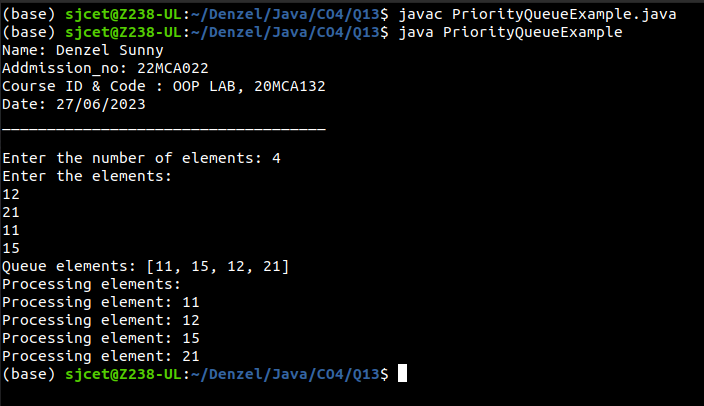
}

scanner.close();

}

}

**Output:**



1. **Program to demonstrate the addition and deletion of elements in deque**

**Code:**

import java.util.Deque;

import java.util.LinkedList;

import java.util.Scanner;

public class DequeExample {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Deque<String> deque = new LinkedList<>();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements to add: ");

int numElementsToAdd = scanner.nextInt();

System.out.println("Enter the elements to add:");

for (int i = 0; i < numElementsToAdd; i++) {

String element = scanner.next();

deque.addLast(element);

}

System.out.println("Deque elements after addition: " + deque);

System.out.print("Enter the number of elements to remove: ");

int numElementsToRemove = scanner.nextInt();

System.out.println("Removed elements:");

for (int i = 0; i < numElementsToRemove; i++) {

String removedElement = deque.removeFirst();

System.out.println(removedElement);

}

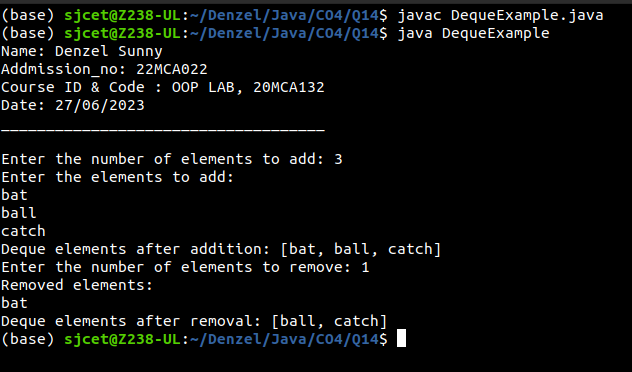
System.out.println("Deque elements after removal: " + deque);

scanner.close();

}

}

**Output:**



1. **Program to demonstrate the creation of Set object using the LinkedHashset class**

**Code:**

import java.util.LinkedHashSet;

import java.util.Scanner;

import java.util.Set;

public class LinkedHashSetexample {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Set<String> set = new LinkedHashSet<>();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements to add: ");

int numElementsToAdd = scanner.nextInt();

System.out.println("Enter the elements to add:");

for (int i = 0; i < numElementsToAdd; i++) {

String element = scanner.next();

set.add(element);

}

System.out.println("Set elements after addition: " + set);

System.out.print("Enter an element to remove: ");

String elementToRemove = scanner.next();

set.remove(elementToRemove);

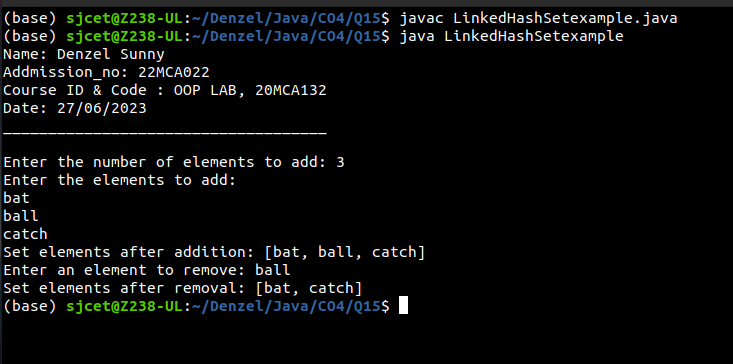
System.out.println("Set elements after removal: " + set);

scanner.close();

}

}

**Output:**



1. **Write a Java program to compare two hash set**

**Code:**

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

public class HashSetComparison {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Set<String> set1 = new HashSet<>();

Set<String> set2 = new HashSet<>();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements for the first set: ");

int numElements1 = scanner.nextInt();

System.out.println("Enter the elements for the first set:");

for (int i = 0; i < numElements1; i++) {

String element = scanner.next();

set1.add(element);

}

System.out.print("Enter the number of elements for the second set: ");

int numElements2 = scanner.nextInt();

System.out.println("Enter the elements for the second set:");

for (int i = 0; i < numElements2; i++) {

String element = scanner.next();

set2.add(element);

}

boolean isEqual = set1.equals(set2);

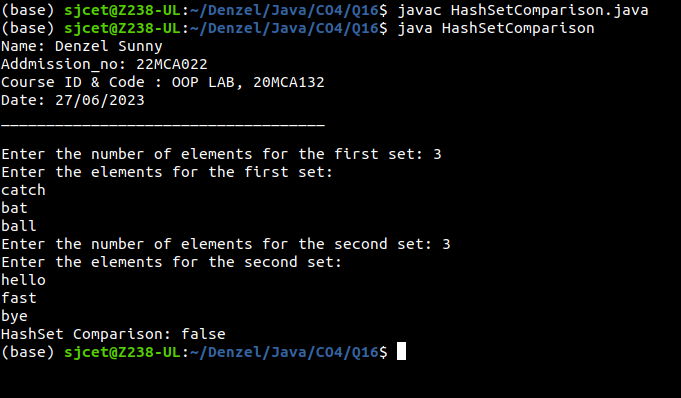
System.out.println("HashSet Comparison: " + isEqual);

scanner.close();

}

}

**Output:**



1. **Program to demonstrate the working of Map interface by adding, changing and removing elements.**

**Code:**

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class MapInterfaceExample {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Map<String, Integer> map = new HashMap<>();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements to add: ");

int numElementsToAdd = scanner.nextInt();

System.out.println("Enter the elements (key-value pairs) to add:");

for (int i = 0; i < numElementsToAdd; i++) {

String key = scanner.next();

int value = scanner.nextInt();

map.put(key, value);

}

System.out.println("Map elements after addition: " + map);

System.out.print("Enter a key to change its value: ");

String keyToChange = scanner.next();

System.out.print("Enter the new value: ");

int newValue = scanner.nextInt();

map.put(keyToChange, newValue);

System.out.println("Map elements after changing value: " + map);

System.out.print("Enter a key to remove its element: ");

String keyToRemove = scanner.next();

map.remove(keyToRemove);

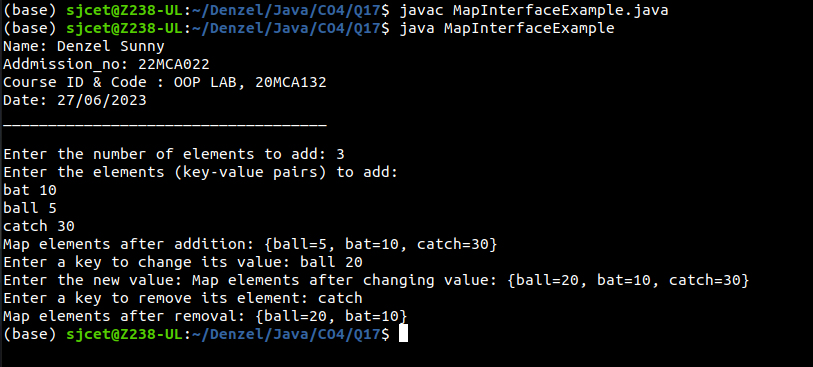
System.out.println("Map elements after removal: " + map);

scanner.close();

}

}

**Output:**



1. **Program to Convert HashMap to TreeMap**

**Code:**

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

import java.util.TreeMap;

public class HashToTree {

public static void main(String[] args) {

System.out.println("Name: Denzel Sunny");

System.out.println("Addmission\_no: 22MCA022");

System.out.println("Course ID & Code : OOP LAB, 20MCA132");

System.out.println("Date: 27/06/2023");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

Map<String, Integer> hashMap = new HashMap<>();

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of elements to add: ");

int numElementsToAdd = scanner.nextInt();

System.out.println("Enter the elements (key-value pairs) to add:");

for (int i = 0; i < numElementsToAdd; i++) {

String key = scanner.next();

int value = scanner.nextInt();

hashMap.put(key, value);

}

System.out.println("HashMap elements: " + hashMap);

Map<String, Integer> treeMap = new TreeMap<>(hashMap);

System.out.println("TreeMap elements: " + treeMap);

scanner.close();

}

}

**Output:**

