**OBJECT ORIENTED PROGRAMMING LAB**

**Name: Sanio Luke Sebastian**

**Roll No: 35**

**Batch: B**

**Date: 31-05-2022**

**Lab Cycle No.: 4**

**Aim**

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.
2. Write a user defined exception class to authenticate the user name and password.
3. Find the average of N positive integers, raising a user defined exception for each negative input.
4. Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).
5. Program to create a generic stack and do the Push and Pop operations.
6. Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.
7. Program to demonstrate the creation of queue object using the Priority Queue class.
8. Program to demonstrate the addition and deletion of elements in dequeue.
9. Write a Java program to compare two hash set.
10. Program to demonstrate the working of Map interface by adding, changing and removing elements.

**Procedure & Outputs**

1. **ques01.java**

import java.util.\*;

import Graphics.\*;

public class ques01 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int choice,isexit=0;

while(isexit==0){

double length, breadth, side, radius;

System.out.println("\n1. Area of Triangle.\n2. Area of Circle.\n3. Area of Square.\n4. Area of Rectangle.\n5. Exit");

System.out.print("Please enter the operation choice to perform - ");

choice= sc.nextInt();

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

switch(choice){

case 1:{

System.out.print("Enter the length of the triangle : ");

length= sc.nextDouble();

System.out.print("Enter the height of the triangle : ");

breadth= sc.nextDouble();

Triangle triangle = new Triangle(length, breadth);

triangle.Area();

break;

}

case 2:{

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

radius= sc.nextDouble();

Circle cir= new Circle(radius);

cir.Area();

break;

}

case 3:{

System.out.print("Enter the side length of the square : ");

side= sc.nextDouble();

Square square= new Square(side);

square.Area();

break;

}

case 4:{

System.out.print("Enter the length of the rectangle : ");

length= sc.nextDouble();

System.out.print("Enter the breadth of the rectangle : ");

breadth= sc.nextDouble();

Rectangle rec= new Rectangle(length, breadth);

rec.Area();

break;

}

case 5:{

isexit=1;

break;

}

default:{

break;

}

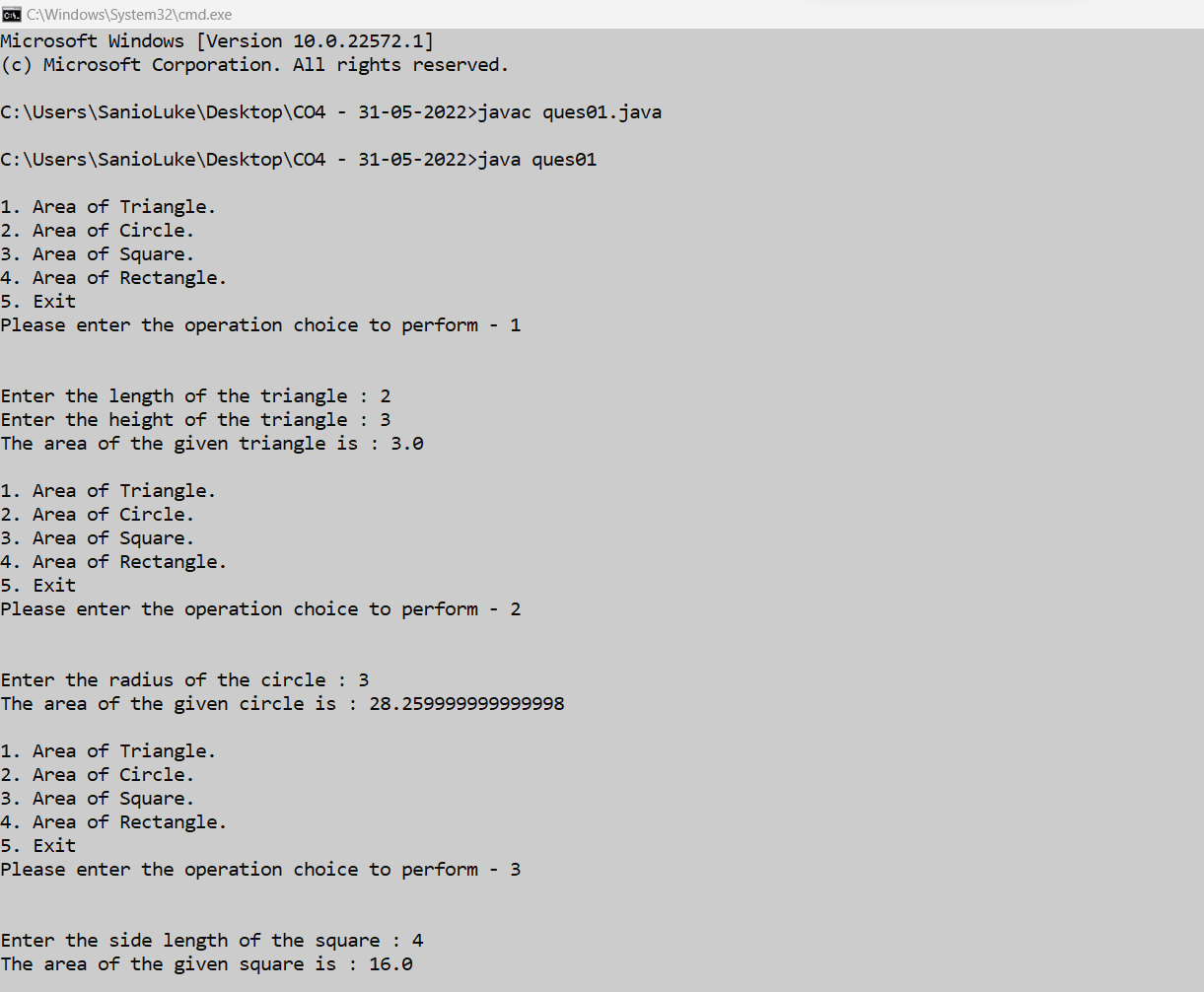
}

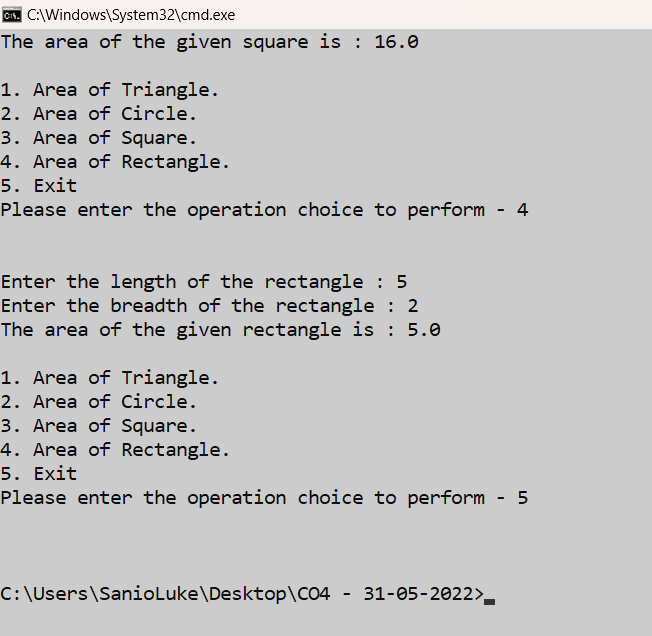
}

sc.close();

}

}





1. **ques02.java**

import java.util.Scanner;

class CustomFormException extends Exception{

public CustomFormException(String errorMessage){

super(errorMessage);

}

}

public class ques02 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

String username, password;

String passregex = "^(?=.\*[0-9])"

+ "(?=.\*[a-z])(?=.\*[A-Z])"

+ "(?=.\*[@#$%^&+=])"

+ "(?=\\S+$).{8,20}$";

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

username= sc.nextLine();

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

password= sc.nextLine();

try {

if(username==null){

sc.close();

throw new CustomFormException("The Username is empty !! Cannot be empty !! Please fill.");

}

else if(password==null){

sc.close();

throw new CustomFormException("The password is empty !! Cannot be empty !! Please fill.");

}

else if(username.trim().isEmpty()){

sc.close();

throw new CustomFormException("Only white spaces allowed in username is not allowed !!");

}

else if(password.trim().isEmpty()){

sc.close();

throw new CustomFormException("Only white spaces allowed in pass is not allowed !!");

}

else if(username.length() < 4){

sc.close();

throw new CustomFormException("The username should be greater than 4 characters !!");

}

else if(!password.matches(passregex)){

sc.close();

throw new CustomFormException("The password does match the pattern !!\nThe password should have atleast-\n1. One lower case alphabet.\n2. One upper case alphabet.\n3. One numeric value.\n4. One special characters like !@#$%^&\*\_ etc.\n5. 8-20 characters.");

}

else {

System.out.println("Form Submitted Successfully.");

}

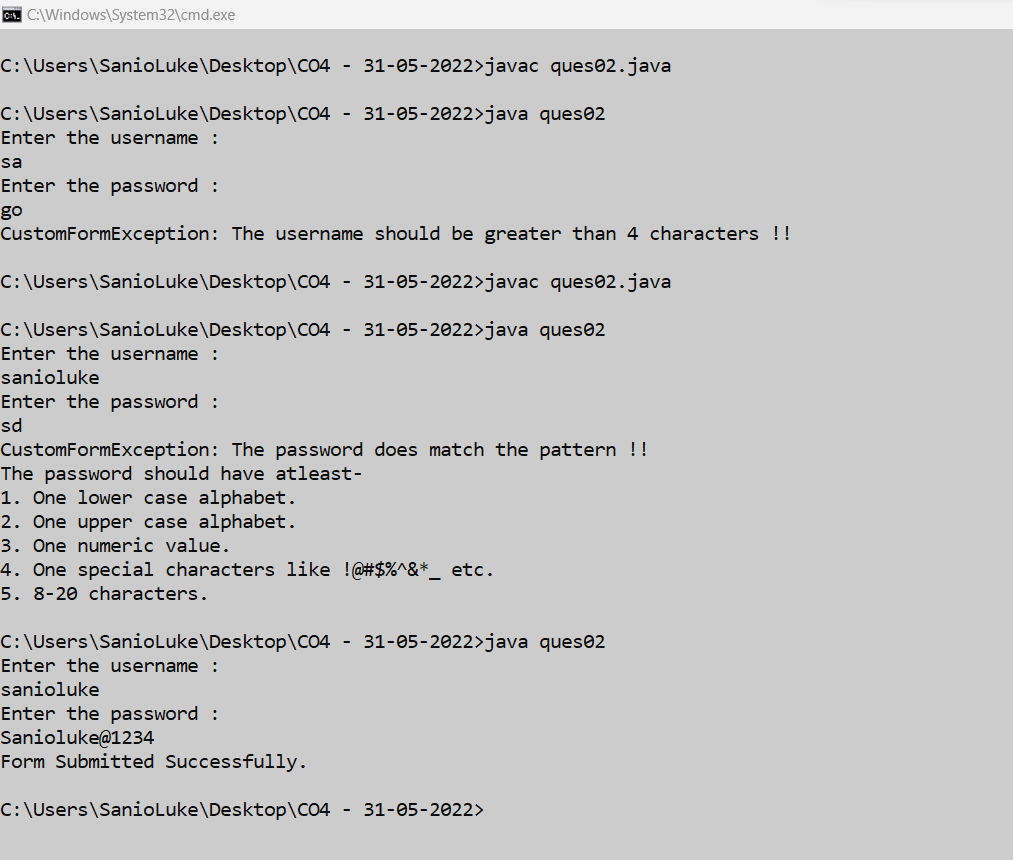
} catch (Exception e) {

System.out.println(e);

}

}

}



1. **ques03.java**

import java.util.Scanner;

class NegativeNumberException extends Exception{

public NegativeNumberException(String error){

super(error);

}

}

public class ques03 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

System.out.print("Enter the number of integers you want to enter : ");

int size= sc.nextInt();

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

int arr[]= new int[size];

int i=0;

while(i < size){

int num;

System.out.print("Enter the number "+(i+1)+" : ");

num= sc.nextInt();

try{

if(num < 0){

sc.close();

throw new NegativeNumberException("No negative number allowed !! Please try again !!");

}

else{

arr[i]= num;

i++;

}

}

catch(Exception e){

System.out.println(e);

}

}

int sum=0;

System.out.print("The elements of the array are : [");

for(i=0;i<size;i++){

System.out.print(arr[i]+", ");

sum+=arr[i];

}

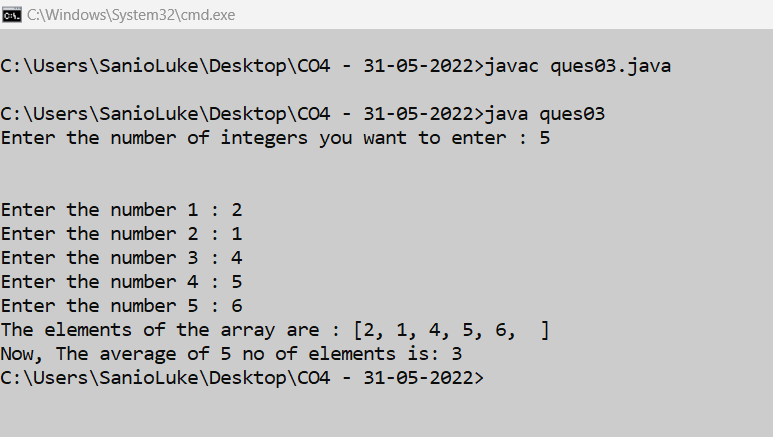
System.out.print(" ]\n");

System.out.print("Now, The average of "+size+" no of elements is: "+(sum / size));

sc.close();

}

}



1. **ques04.java**

import java.util.Scanner;

class Fibonacci extends Thread{

int size;

Fibonacci(int size){

this.size=size;

}

public void run(){

int num1=0, num2=1;

System.out.println("Fibonacci - 0, 1, ");

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

int temp=num1;

num1= num2;

num2= temp+num1;

System.out.println("Fibnacci - "+num2+", ");

}

}

}

class EvenNumber extends Thread{

int range;

EvenNumber(int range){

this.range= range;

}

public void run(){

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

if(i%2==0){

System.out.println("The even number : "+i);

}

}

}

}

public class ques04 {

public static void main(String[] args) {

int size, range;

Scanner sc= new Scanner(System.in);

System.out.print("Enter the size of fibonacci series : ");

size= sc.nextInt();

System.out.println("Enter the range of even numbers : ");

range= sc.nextInt();

Fibonacci fib= new Fibonacci(size);

EvenNumber even= new EvenNumber(range);

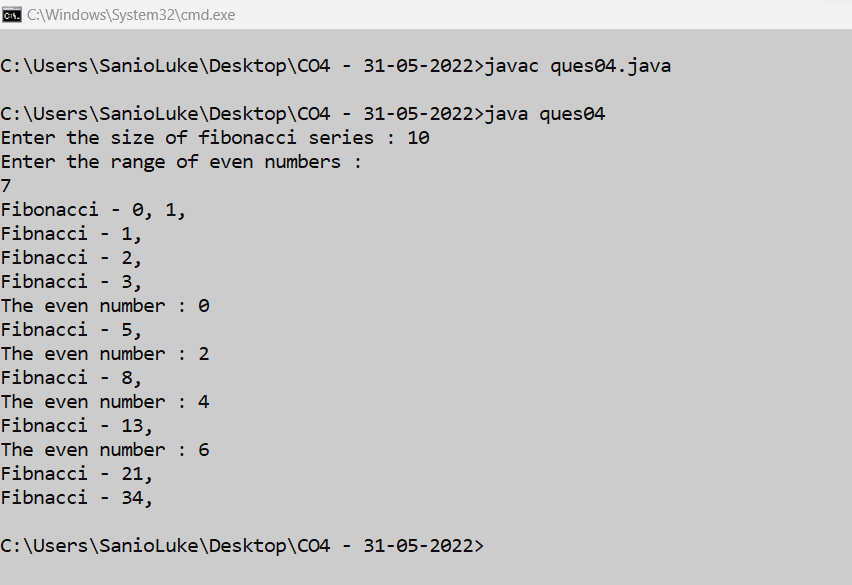
fib.start();

even.start();

sc.close();

}

}



1. **ques05.java**

import java.util.Scanner;

public class ques05 {

public static void main(String[] args) {

int choice, isexit=0;

Scanner sc= new Scanner(System.in);

int stack\_size, top=-1;

System.out.print("\nEnter the size of the stack : ");

stack\_size= sc.nextInt();

int[] a= new int[stack\_size];

while(isexit==0){

System.out.println("\nSelect from the following stack operations that you want to perform :\n1. Push an element.\n2. Pop an element.\n3. Display the stack.\n4. Exit");

System.out.print("\nEnter the choice: ");

choice= sc.nextInt();

switch(choice){

case 1:{

int new\_element;

System.out.print("\nEnter the new element you want to add : ");

new\_element= sc.nextInt();

if(top >= stack\_size-1){

System.out.println("\nStack Overflow !! Cannot push anymore elements; Stack is full !!");

}

else{

a[++top]=new\_element;

}

break;

}

case 2:{

if(top==-1){

System.out.println("Stack underflow !! Cannot pop anymore elements; Stack is empty !!");

}

else{

int pop\_element= a[top--];

System.out.println("\nThe popped element is : "+pop\_element);

}

break;

}

case 3:{

if(top < 0){

System.out.println("Stack underflow !! Cannot display any elements; Stack is empty !!");

}

else{

System.out.print("\nThe elememts in the stack are : [");

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

if(i==top)

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

else

System.out.print(a[i]+", ");

}

System.out.print("]");

}

break;

}

case 4:{

isexit=1;

break;

}

default:{

System.out.println("\nInvalid choice. Please try again !!");

}

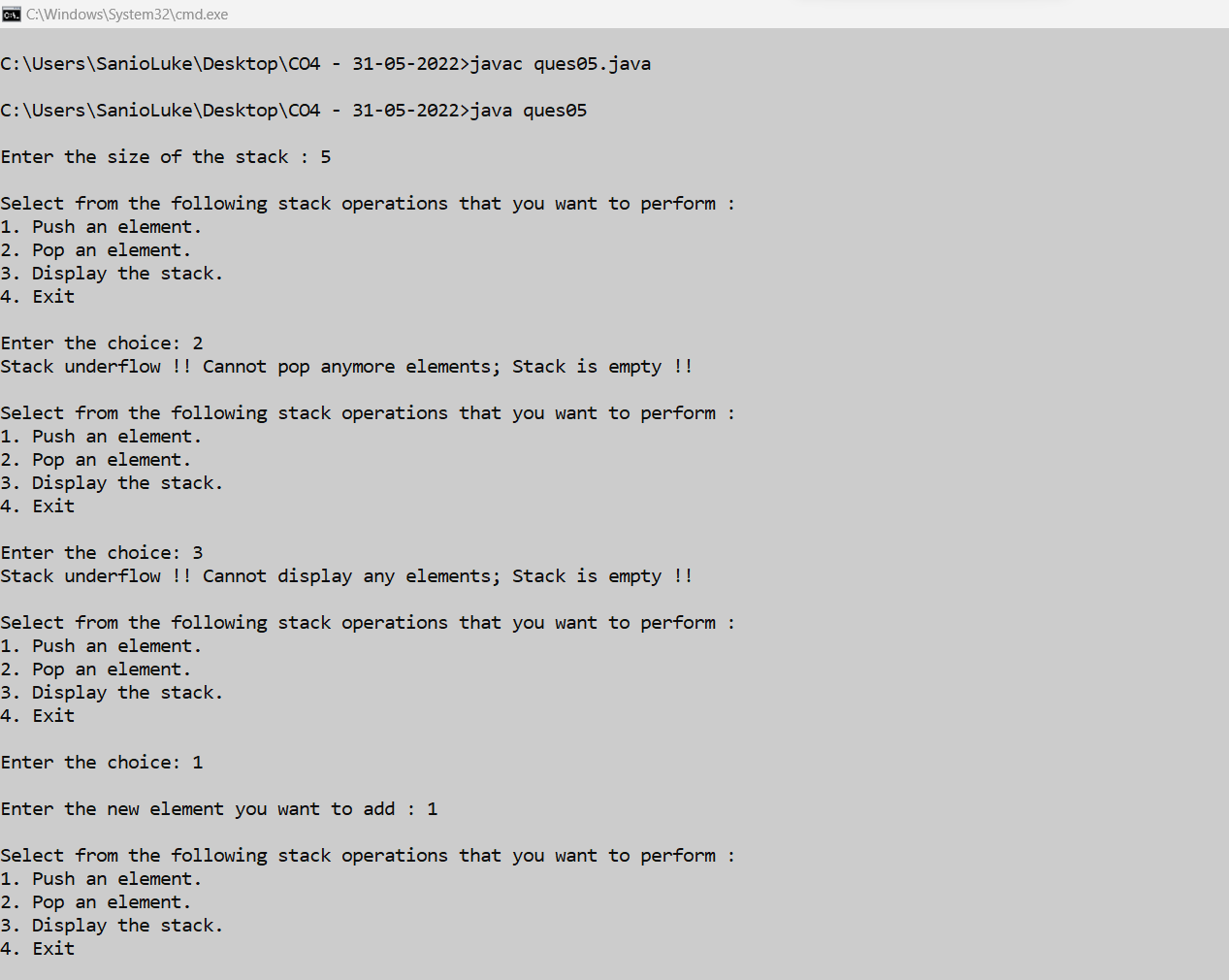
}

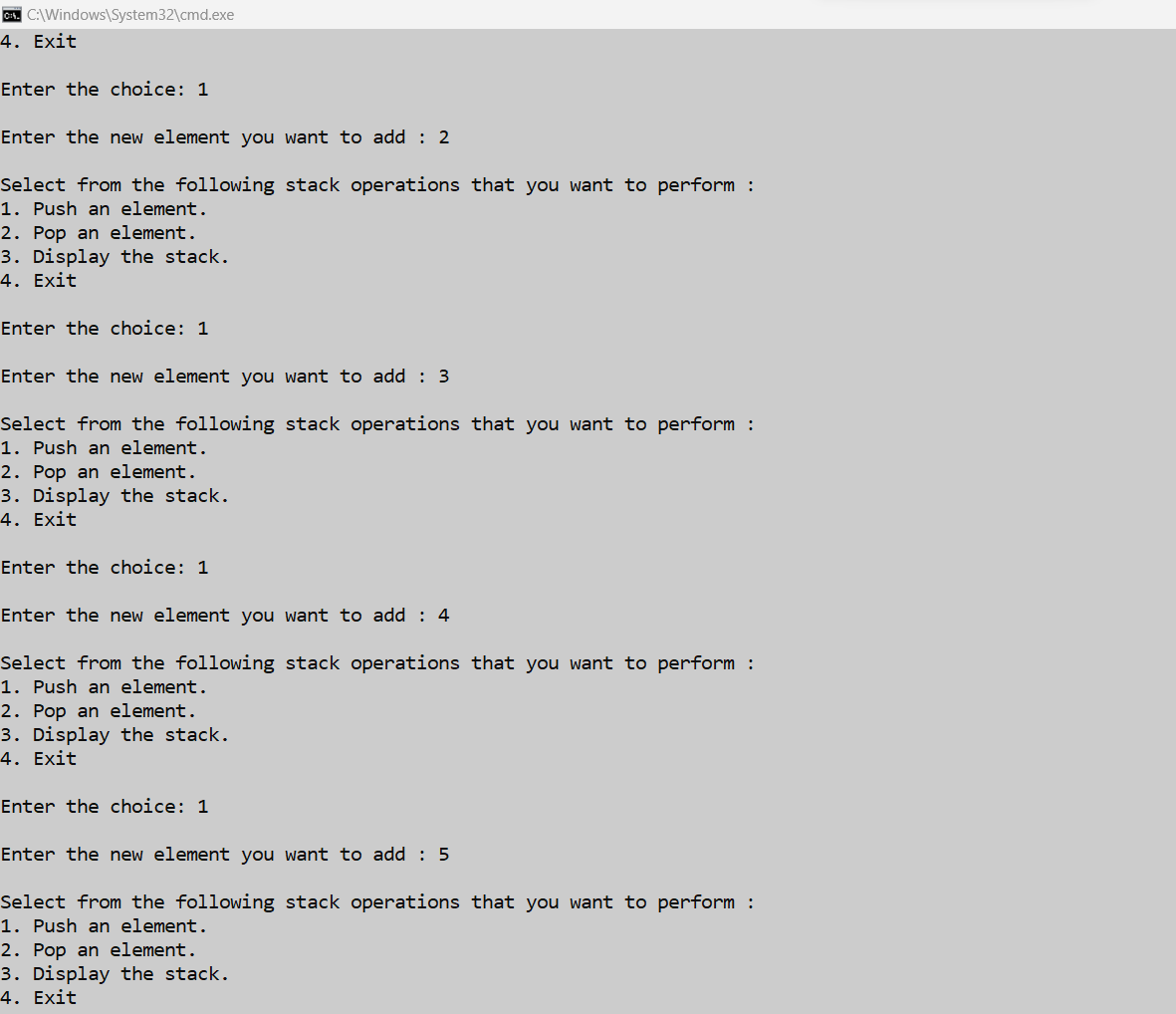
}

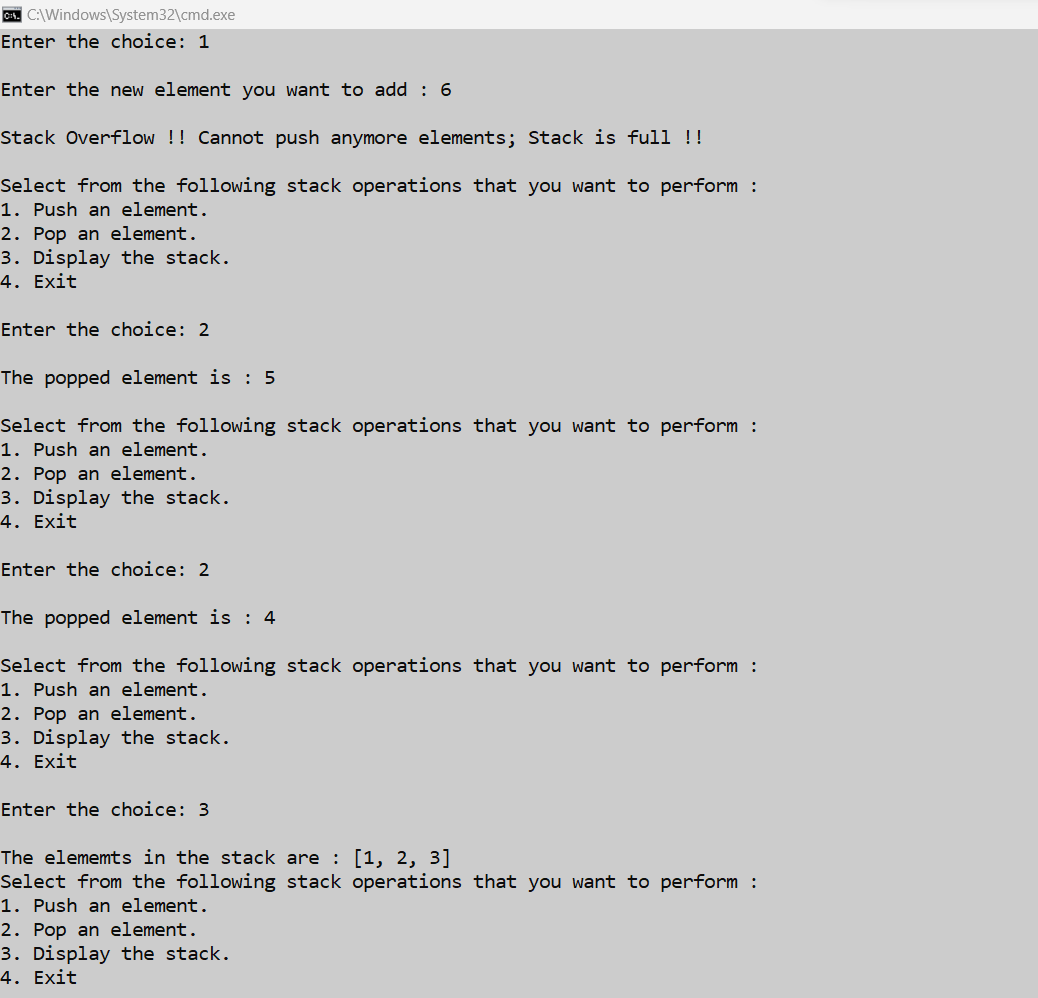
sc.close();

}

}







1. **ques06.java**

import java.util.\*;

public class ques06 {

public static void main(String[] args) {

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

arrayList.add("Avil");

arrayList.add("Sanio");

arrayList.add("Tejas");

arrayList.add("Nebin");

System.out.println("The elements of the arraylist is - "+arrayList);

Collections.sort(arrayList);

System.out.println("\nThe ArrayList Sort : "+arrayList); // ArrayList Sort

Collections.addAll(arrayList,"Vivek","Vikram","Shantanu","Winston","Godwin");

System.out.println("\nAdding new items in the arraylist is : "+arrayList); // ArrayList AddAll

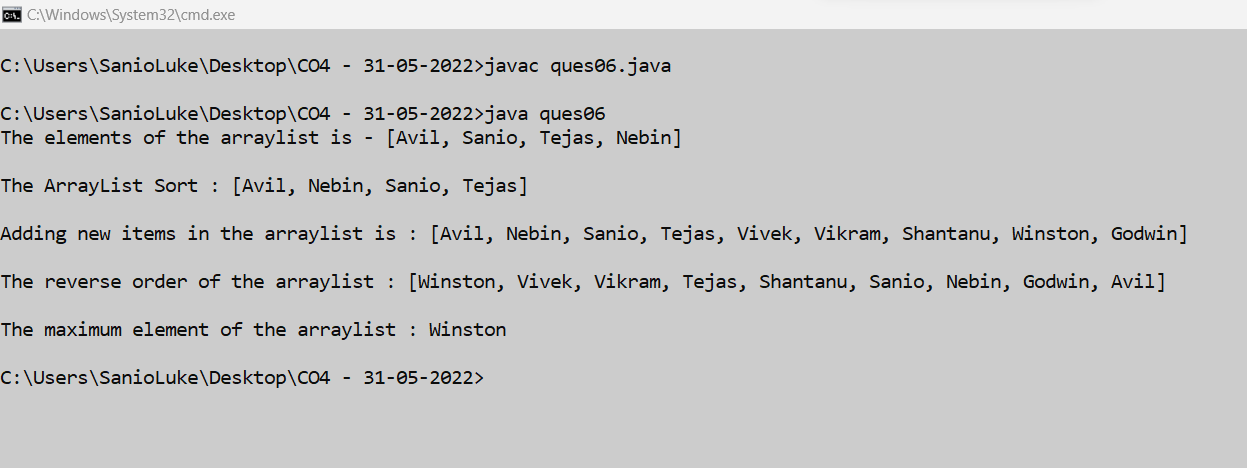
Collections.sort(arrayList, Collections.reverseOrder()); //Arraylist in reverse order

System.out.println("\nThe reverse order of the arraylist : "+arrayList);

System.out.println("\nThe maximum element of the arraylist : "+Collections.max(arrayList)); //Max elements in the arraylist

}

}



1. **ques07.java**

import java.util.\*;

public class ques07 {

public static void main(String[] args) {

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

Scanner sc= new Scanner(System.in);

boolean iscontinue= true;

int choice;

while(iscontinue){

System.out.println("\nFollowing are the operations that you can perform on a PriorityQueue:\n1. Insertion of an element.\n2. Deletion of an element.\n3. Display the top element (peek).\n4. Exit");

System.out.print("Select your choice: ");

choice= sc.nextInt();

switch(choice){

case 1: {

System.out.print("\nEnter the element that you want to insert: ");

queue.add(sc.nextInt());

break;

}

case 2: {

if(queue.size() <= 0)

System.out.println("\nCannot delete from the queue !! Queue is empty !!");

else

System.out.println("The deleted element is : "+queue.poll());

break;

}

case 3: {

if(queue.size() <= 0)

System.out.println("\nCannot delete from the queue !! Queue is empty !!");

else

System.out.println("The top element (peek) is : "+queue.peek());

break;

}

case 4: {

iscontinue= false;

break;

}

default:{

System.out.println("\nInvalid choice !! Please try again !!");

}

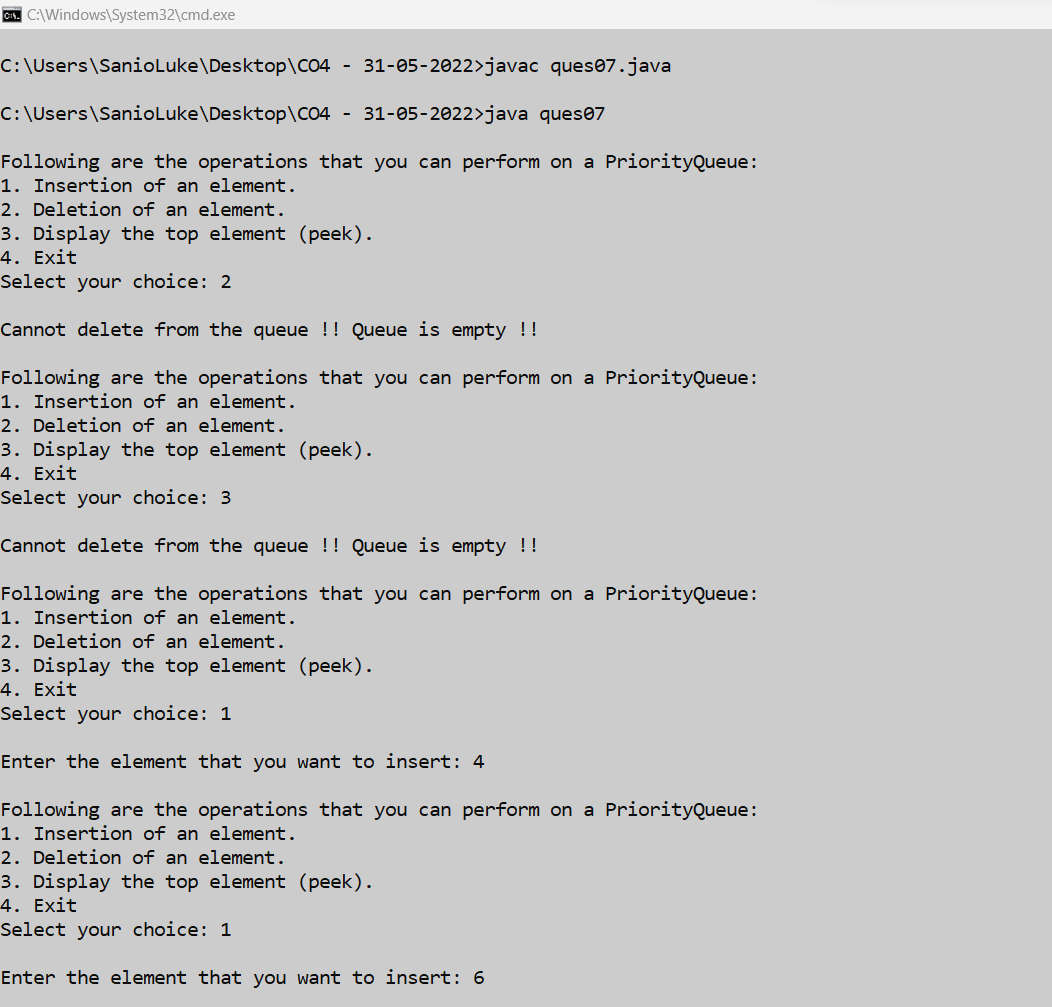
}

}

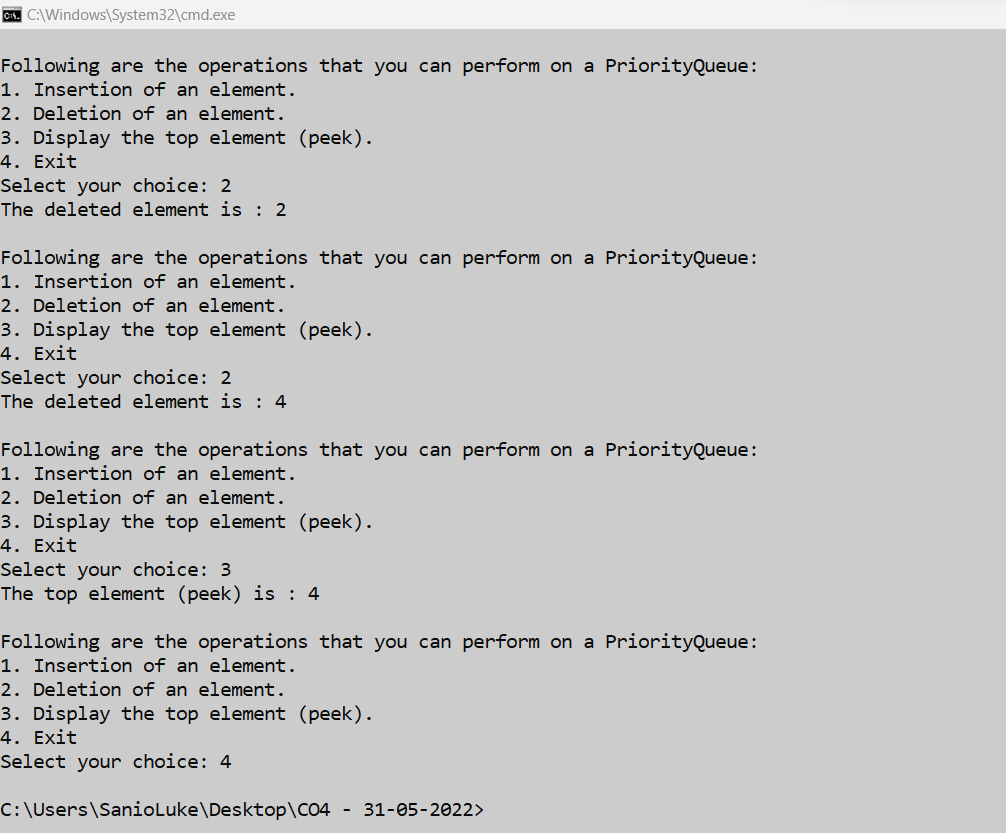
sc.close();

}

}







1. **ques08.java**

import java.util.\*;

public class ques08 {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

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

// Add at the last

deque.add("Item 1 - Tail");

// Add at the first

deque.addFirst("Item 2 - Head");

// Add at the last

deque.addLast("Item 3 - Tail");

// Add at the first

deque.push("Item 4 - Head");

// Add at the last

deque.offer("Item 5 - Tail");

// Add at the first

deque.offerFirst("Item 6 - Head");

System.out.println(deque + "\n");

// We can remove the first element or the last element.

deque.removeFirst();

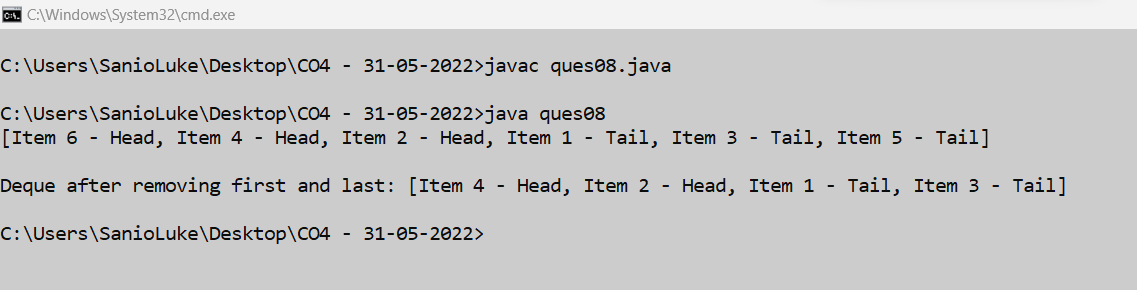
deque.removeLast();

System.out.println("Deque after removing first and last: "+deque);

sc.close();

}

}



1. **ques09.java**

import java.util.\*;

public abstract class ques09 {

public static void main(String[] args) {

HashSet<Integer> hashset\_1= new HashSet<>();

hashset\_1.add(13);

hashset\_1.add(5);

hashset\_1.add(53);

hashset\_1.add(47);

hashset\_1.add(63);

System.out.println("\nHashset 01 : ");

Iterator<Integer> i1=hashset\_1.iterator();

while(i1.hasNext())

System.out.println(i1.next());

HashSet<Integer> hashset\_2= new HashSet<>();

hashset\_2.add(17);

hashset\_2.add(5);

hashset\_2.add(2);

hashset\_2.add(47);

hashset\_2.add(3);

System.out.println("\nHashset 02 : ");

Iterator<Integer> i2=hashset\_2.iterator();

while(i2.hasNext())

System.out.println(i2.next());

System.out.println("\nElements that are common in both of the hashset are : ");

for(Integer elem: hashset\_1){

if(hashset\_2.contains(elem))

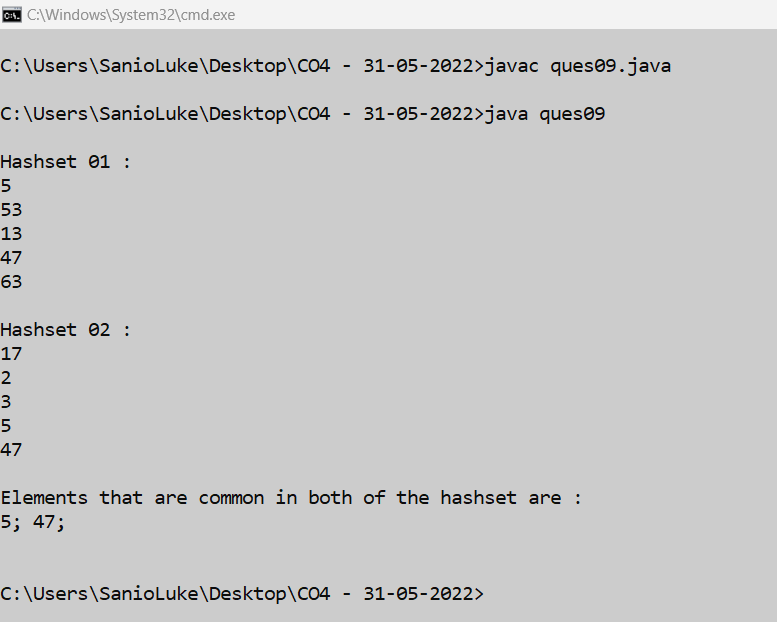
System.out.print(elem+"; ");

}

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

}

}



1. **ques10.java**

import java.util.\*;

public class ques10 {

public static void main(String[] args) {

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

Scanner sc= new Scanner(System.in);

int choice;

boolean iscontinue=true;

int map\_count=0;

while(iscontinue){

System.out.println("\nMap Operations that you can perform are:\n1. Adding an element.\n2. Changing an element.\n3. Deleting an element.\n4. Display all elements.\n5. Exit.");

System.out.print("\nEnter the choice: ");

choice= sc.nextInt();

switch(choice){

case 1: {

System.out.print("\nEnter the string value for new element: ");

String item= sc.nextLine()+ sc.nextLine();

map\_count++;

map.put(map\_count,item);

break;

}

case 2: {

if(map.size() <= 0){

System.out.println("\nCannot chnage any element !! The map is empty.");

}

else{

System.out.print("\nEnter the key value of the element that you want to change: ");

int key= sc.nextInt();

boolean iskeyvalid= false;

for(Map.Entry<Integer, String> item: map.entrySet()){

if(key==item.getKey()){

System.out.print("\nEnter the new element to update from the map: ");

String value= sc.nextLine()+ sc.nextLine();

map.put(key,value);

iskeyvalid= true;

break;

}

}

if(!iskeyvalid){

System.out.println("\nInvalid key value !! Please enter an existing key value.");

}

}

break;

}

case 3: {

if(map.size() <= 0){

System.out.println("\nCannot delete any element !! The map is empty.");

}

else{

System.out.print("\nEnter the key value of the element that you want to delete: ");

int key= sc.nextInt();

boolean iskeyvalid= false;

for(Map.Entry<Integer, String> item: map.entrySet()){

if(key==item.getKey()){

iskeyvalid=true;

map.remove(key);

System.out.println("\nThe mentioned element is successfully deleted.");

break;

}

}

if(!iskeyvalid)

System.out.println("\nInvalid key value !! Please enter an existing key value.");

}

break;

}

case 4: {

if(map.size() <= 0){

System.out.println("\nCannot display any elements !! The map is empty.");

}

else{

System.out.println("\nThe elements in the map are : ");

for(Map.Entry<Integer, String> item: map.entrySet())

System.out.println("\nItem Key- "+item.getKey()+" & Item Value- "+item.getValue());

}

break;

}

case 5: {

iscontinue= false;

break;

}

default:{

System.out.println("\nInvalid choice !! Please try again !!");

}

}

}

sc.close();

}

}

