HINDUSTHAN COLLEGE OF ARTS & SCIENCE (AUTONOMOUS)

An Autonomous Institution - Affiliated to Bharathiar University

(ISO 9001-2001 Certified Institution)

Behind Nava India, Coimbatore – 641028.

PG AND RESEARCH DEPARTMENT OF COMPUTER APPLICATIONS

MASTER OF COMPUTER APPLICATOINS

PRACTICAL RECORD

25MCPP08 – PRACTICAL: PROGRAMMING IN JAVA WITH DATA STRUCTURES

NAME : _____

REGISTER NO : _____

CLASS : _____

SEMESTER : _____

YEAR : _____

HINDUSTHAN COLLEGE OF ARTS & SCIENCE (AUTONOMOUS)

An Autonomous Institution - Affiliated to Bharathiar University

(ISO 9001-2001 Certified Institution)

Behind Nava India, Coimbatore – 641028.

PG AND RESEARCH DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

Certified that this is a bonafide record of **PROGRAMING IN JAVA WITH DATA STRUCTURES (25MCPP08)** done by_____

Register No:_____during the academic year 2025-2026.

STAFF IN – CHARGE **HEAD OF THE DEPARTMENT**

Submitted for the Bharathiar University practical Examination held on _____

at Hindusthan College of Arts & Science, Coimbatore - 28.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Date:

Place: Coimbatore

CONTENTS

EX.NO: 01 DATE:	IMPLEMENTATION OF INHERITANCE	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
// Parent class
class Animal {
void eat() {
System.out.println("The animal is eating");
}
void sleep() {
System.out.println("The animal is sleeping");
}
}
// Child class inheriting from Animal
class Dog extends Animal {
void bark() {
System.out.println("The dog is barking");
}
}
// Another child class inheriting from Animal
class Cat extends Animal {
void meow() {
System.out.println("The cat is meowing");
}
}
// Main class to test the inheritance
public class InheritanceExample {
public static void main(String[] args) {
// Creating objects of the child classes
Dog myDog = new Dog();
Cat myCat = new Cat();
// Calling methods from the parent class
myDog.eat();
```

```
myDog.sleep();
// Calling methods from the child class
myDog.bark();
// Calling methods from another child class
myCat.eat();
myCat.sleep();
myCat.meow();
}
}
```

OUTPUT:

```
The animal is eating
The animal is sleeping
The dog is barking
The animal is eating
The animal is sleeping
The cat is meowing
```

RESULT:

EX.NO: 02 DATE:	IMPLEMENTATION OF ARRAY	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
import java.util.Scanner;
public class Insert_Array
{
public static void main(String[] args)
{
int n, pos, x;
Scanner s = new Scanner(System.in);
System.out.print("Enter no. of elements you want in array:");
n = s.nextInt();
int a[] = new int[n+1];
System.out.println("Enter all the elements:");
for(int i = 0; i < n; i++)
{
a[i] = s.nextInt();
}
System.out.print("Enter the position where you want to insert element:");
pos = s.nextInt();
System.out.print("Enter the element you want to insert:");
x = s.nextInt();
for(int i = (n-1); i >= (pos-1); i--)
{
a[i+1] = a[i];
}
a[pos-1] = x;

System.out.print("After inserting:");
for(int i = 0; i < n; i++)
{

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

```
}
System.out.print(a[n]);
}
}
```

OUTPUT:

```
Enter no. of elements you want in array:5
Enter all the elements:
8
9
3
5
7
Enter the position where you want to insert element:2
Enter the element you want to insert:6
After inserting:8,6,9,3,5,7
```

RESULT:

EX.NO: 03 DATE:	PALINDROME CHECKING	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
import java.util.Scanner;
public class palindrome {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a string to check if it is a palindrome: ");
String s = scanner.nextLine();
String rev = "";
for (int i = s.length()-1; i >=0 ; i--)
rev=rev+s.charAt(i);
if(s.equals(rev))
System.out.println("The Given String "+s+" is palindrome");
else
System.out.println("The Given String "+s+" is not palindrome");
}
}
```

OUTPUT:

```
Enter a string to check if it is a palindrome: LEVEL
The Given String LEVEL is palindrome
```

```
Enter a string to check if it is a palindrome: TAMIL
The Given String TAMIL is not palindrome
```

RESULT:

EX.NO: 04 DATE:	HANDLING DIFFERENT MOUSE EVENTS	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

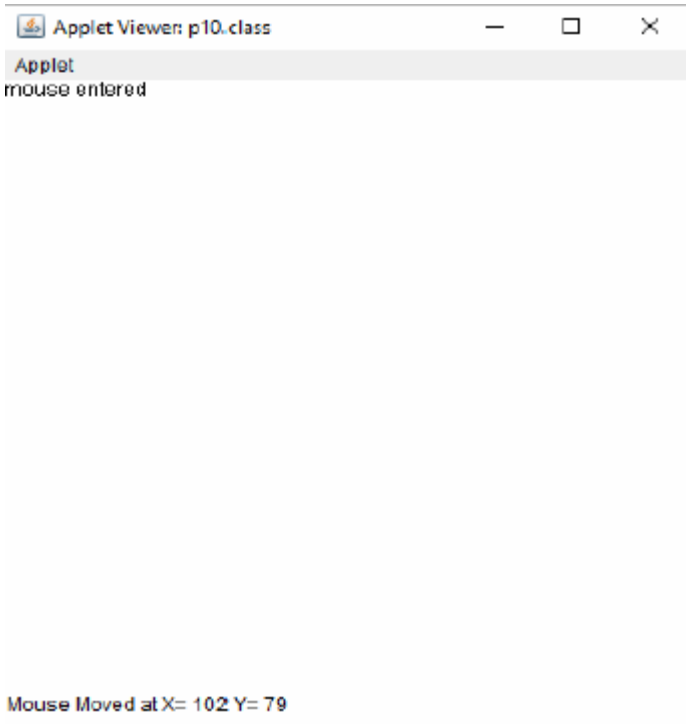
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class pr4 extends Applet implements MouseListener,MouseMotionListener
{
String msg=" ";
int mousex=0;
int mousey=0;
public void init()
{
addMouseListener(this);
addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent e)
{
mousex=0;
mousey=10;
msg="mouse clicked";
repaint();
}
public void mouseEntered(MouseEvent e)
{
mousex=0;
mousey=10;
msg="mouse entered";
repaint();
}
public void mouseExited(MouseEvent e)
{
mousex=0;
mousey=10;
msg="mouse exited";
repaint();
```

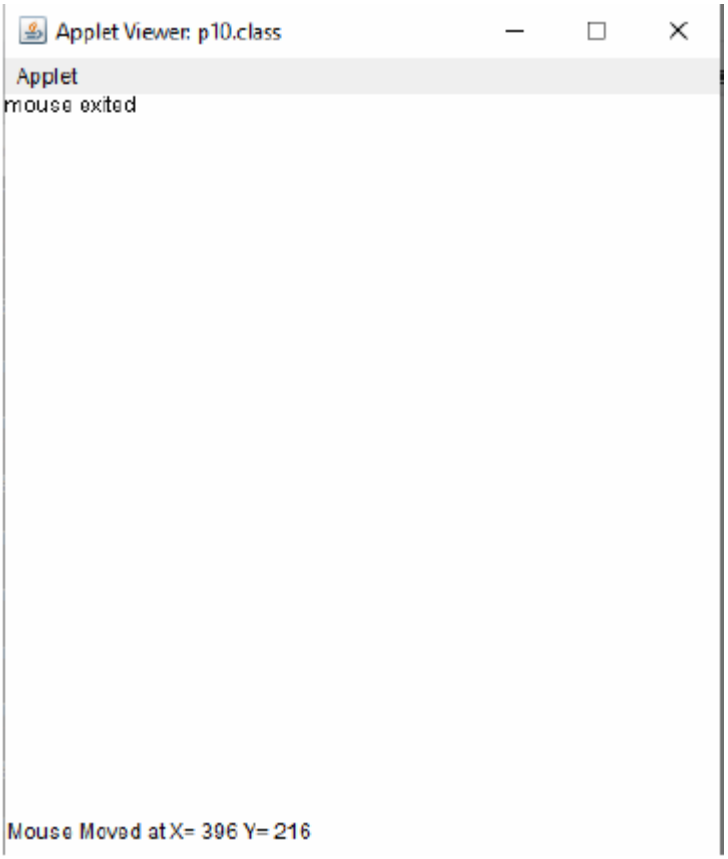
```
}
public void mouseReleased(MouseEvent e)
{
mousex=e.getX();
mousey=e.getY();
msg="up";
repaint();
}
public void mousePressed(MouseEvent e)
{
mousex=e.getX();
mousey=e.getY();
msg="down";
repaint();
}
public void mouseDragged(MouseEvent e)
{
mousex=e.getX();
mousey=e.getY();
msg="mouse dragged";
repaint();
}
public void mouseMoved(MouseEvent e)
{
showStatus("Mouse Moved at X= "+e.getX()+" Y= "+e.getY());
}
public void paint(Graphics g)
{
g.drawString(msg,mousex,mousey);
}
}
```

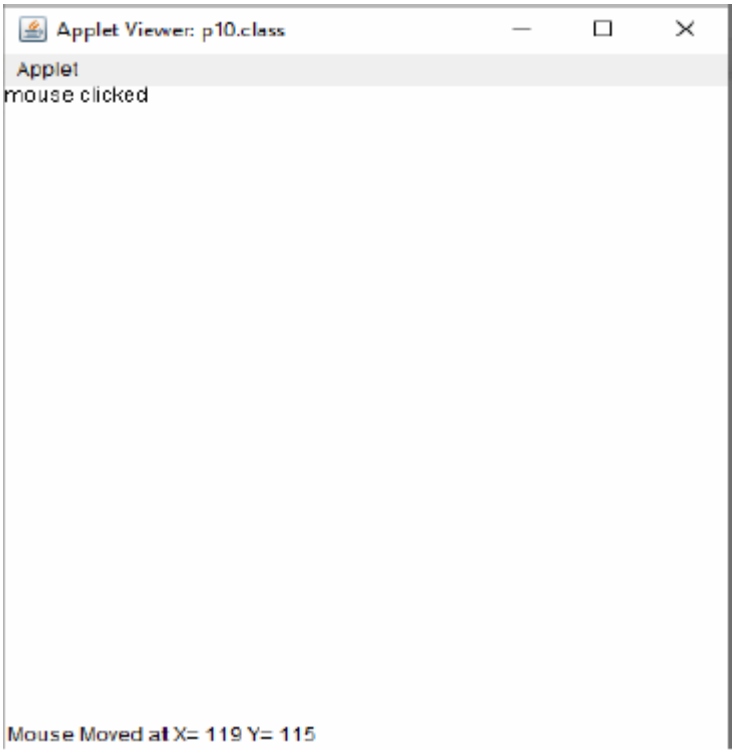
Pr4.html

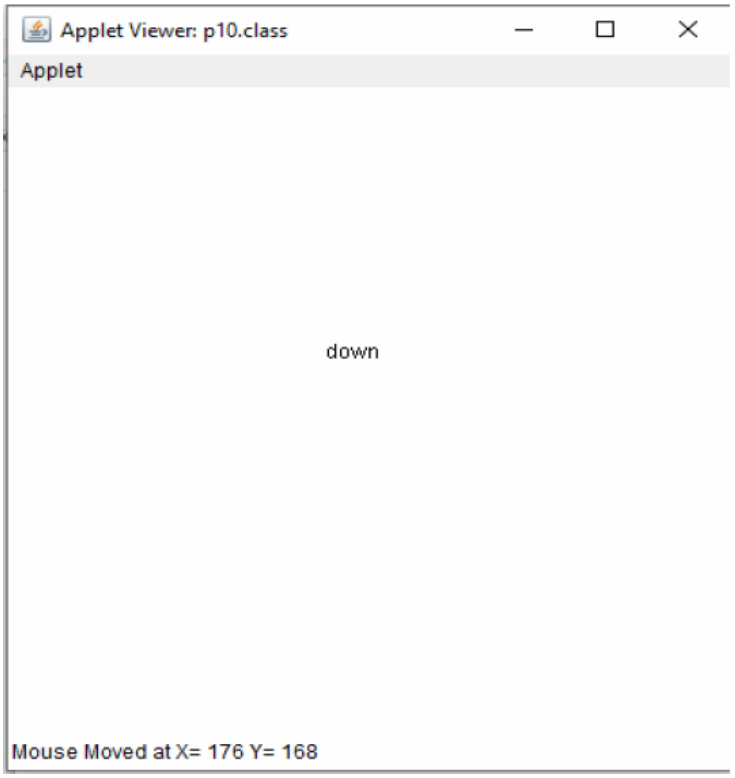
```
<html>
<applet code=pr4.class width=400 height=400>
</applet>
</html>
```

OUTPUT:


Applet Viewer: p10.class
Applet
mouse entered
Mouse Moved at X= 102 Y= 79


Applet Viewer: p10.class
Applet
mouse exited
Mouse Moved at X= 396 Y= 216


Applet Viewer: p10.class
Applet
mouse clicked
Mouse Moved at X= 119 Y= 115


Applet Viewer: p10.class
Applet
down
Mouse Moved at X= 176 Y= 168

RESULT:

EX.NO: 05 DATE:	ASCENDING ORDER	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
import java.util.*;
class sorting
{
void sortStrings()
{
Scanner s=new Scanner(System.in);
System.out.println("Enter the values of n:");
int n=s.nextInt();
String[]str=new String[n];
System.out.println("Enter Values:");
for(int i=0;i<n;i++)
{
str[i]=new String(s.next());
}
for(int i=0;i<n;i++)
{
for(int j=i+1;j<n;j++)
{
if(str[i].compareTo(str[j])>0)
{
String temp=str[i];
str[i]=str[j];
str[j]=temp;
}
}
}
System.out.println("Ascending order of the Given Values:");
for(int i=0;i<n;i++)
{
System.out.println(str[i]);
}
}
}
```

```
class pr5
{
public static void main(String[]args)
{
sorting obj=new sorting();
obj.sortStrings();
}
}
```

OUTPUT:

```
Enter the values of n:
4
Enter Values:
dinesh
prakash
kumar
anand
Ascending order of the Given Values:
anand
dinesh
kumar
prakash
```

RESULT:

EX.NO: 06 DATE:	IMPLEMENTATION OF LINKED LIST	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
import java.util.LinkedList;
class Main {
public static void main(String[] args){
// create linkedlist
LinkedList<String> animals = new LinkedList<>();
// add() method without the index parameter
animals.add("Dog");
animals.add("Cat");
animals.add("Cow");
System.out.println("LinkedList: " + animals);
// add() method with the index parameter
animals.add(1, "Horse");
System.out.println("After insertion Updated LinkedList: " + animals);
// remove elements from index 0
String str = animals.remove(0);
System.out.println("Removed Element: " + str);
System.out.println("After deletion Updated LinkedList: " + animals);
}
}
```

OUTPUT:

```
LinkedList: [Dog, Cat, Cow]
After insertion Updated LinkedList: [Dog, Horse, Cat, Cow]
Removed Element: Dog
After deletion Updated LinkedList: [Horse, Cat, Cow]
```

RESULY:

EX.NO: 07 DATE:	DISPLAYS THE NUMBER OF CHARACTERS, LINES AND WORDS IN A TEXT FILE	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
import java.io.*;
class FileDemo
{
public static void main(String args[])
{
try
{
int lines=0,chars=0,words=0;
int code=0;
FileInputStream fis = new FileInputStream("sample.txt");
while(fis.available()!=0)
{
code = fis.read();
if(code!=10)
chars++;
if(code==32)
words++;
if(code==13)
{
lines++;
words++;
}
}
System.out.println("No.of characters = "+chars);
System.out.println("No.of words = "+(words+1));
System.out.println("No.of lines = "+(lines+1));
fis.close();
}
catch(FileNotFoundException e)
{
System.out.println("Cannot find the specified file...");
}
```

```
catch(IOException i)
{
System.out.println("Cannot read file...");
}
}
}
```

OUTPUT:

Content in sample.txt file is;

Welcome to

Hindusthan

College of

Arts and Science

Output of the above Program is;

```
No.of characters = 49
No.of words = 8
No.of lines = 4
```

RESULT:

EX.NO: 08 DATE:	DISPLAYS A SIMPLE MESSAGE USING APPLET	PAGE NO:

AIM:

ALGORITHM:

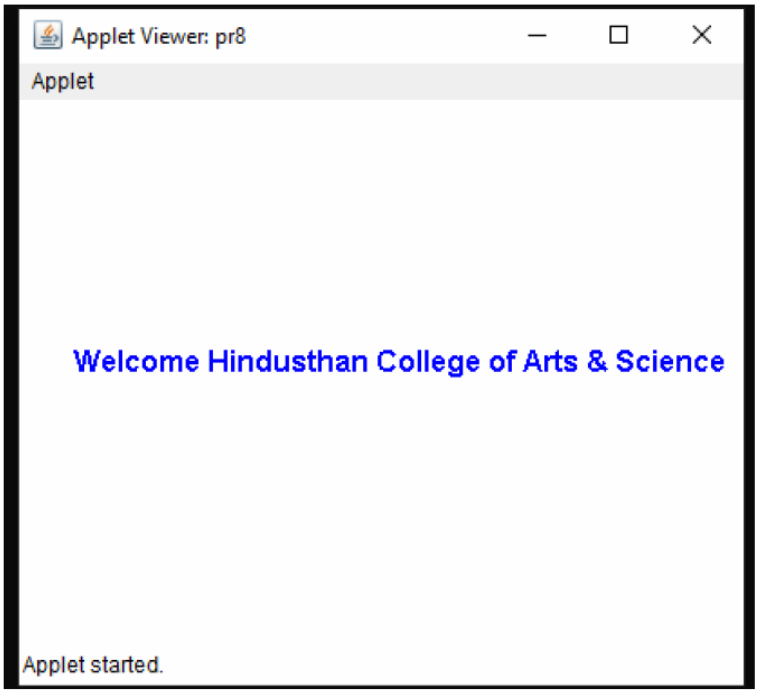
SOURCE CODE:

```
import java.awt.*;
import java.applet.*;
public class pr8 extends Applet
{
public void paint(Graphics g)
{
g.setColor(Color.blue);
Font font = new Font("Arial", Font.BOLD, 16);
g.setFont(font);
g.drawString("Welcome Hindusthan College of Arts & Science",30,150);
}
}
```

Pr8.html

```
<html>
<applet code="pr8" width=400 height=300></applet>
</html>
```

OUTPUT:



RESULT:

EX.NO: 09 DATE:	ACCESS A DATABASE USING JDBC CONNECTION	PAGE NO:

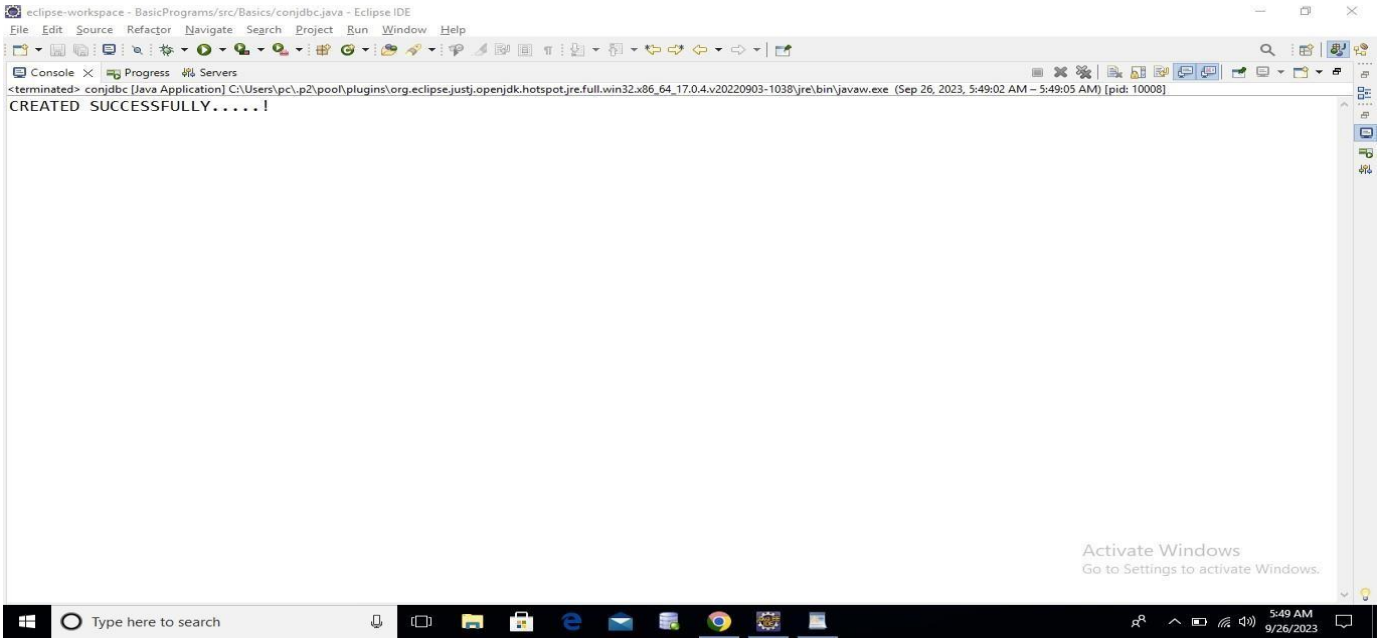
AIM:

ALGORITHM

SOURCE CODE:

```
Importjava.sql.*;
public class conjdbc
{
public static void main(String[] args)
{
String cs;
cs="CREATE TABLE EM12(ENO NUMBER(6),NAME VARCHAR2(25),BPAY
NUMBER(10,2))";
try
{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection c=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","root","root");
if(c!=null)
System.out.println("Connected"); Statement st=c.createStatement(); st.executeUpdate(cs);
st.close();
c.close();
}
catch(ClassNotFoundException e)
{
System.out.println(e.getMessage());
}
catch(SQLException ex) {
System.out.println("SQL EXCEPTION"+ex.getMessage());
}
}
```

OUTPUT:



RESULT:

EX.NO: 10 DATE:	IMPLEMENTATION OF STACKS	PAGE NO:

AIM:

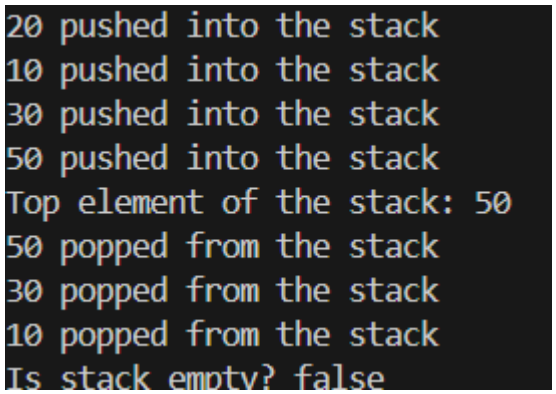
ALGORITHM:

SOURCE CODE:

```
public class pr10 {
private int maxSize;
private int[] stackArray;
private int top;
public pr10(int size) {
maxSize = size;
stackArray = new int[maxSize];
top = -1;
}
// Method to push an element onto the stack
public void push(int value) {
if (top == maxSize - 1) {
System.out.println("Stack overflow");
return;
}
stackArray[++top] = value;
System.out.println(value + " pushed into the stack");
}
// Method to pop an element from the stack
public int pop() {
if (top == -1) {
System.out.println("Stack underflow");
return -1;
}
int poppedElement = stackArray[top--];
System.out.println(poppedElement + " popped from the stack");
return poppedElement;
}
// Method to peek the top element of the stack
public int top() {
if (top == -1) {
System.out.println("Stack is empty");
```

```
return -1;
}
return stackArray[top];
}
// Method to check if the stack is empty
public boolean isEmpty() {
return (top == -1);
}
public static void main(String[] args) {
pr10 stack = new pr10(5); // Creating a stack of size 5
// Pushing elements onto the stack
stack.push(20);
stack.push(10);
stack.push(30);
stack.push(50);
// Peeking the top element
System.out.println("Top element of the stack: " + stack.top());
// Popping elements from the stack
stack.pop();
stack.pop();
stack.pop(); // Trying to pop from an empty stack
// Checking if the stack is empty
System.out.println("Is stack empty? " + stack.isEmpty());
}
}
```

OUTPUT:



RESULT:

EX.NO: 11 DATE:	IMPLEMENTATION OF QUEUES	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
import java.util.Arrays;
public class pr11 {
private int[] arr;
private int front;
private int rear;
private int capacity;
private int size;
public pr11(int capacity) {
this.capacity = capacity;
arr = new int[capacity];
front = 0;
rear = -1;
size = 0;
}
public void enqueue(int element){
if(size == capacity){
System.out.println("Queue is full. Cannot enqueue " + element);
return;
}
rear = (rear + 1) % capacity;
arr[rear] = element;
size++;
System.out.println("Enqueued: " + element);
}
public int dequeue() {
if(size == 0){
System.out.println("Queue is empty. Cannot dequeue.");
return -1;
}
int element = arr[front];
front = (front + 1) % capacity;
size--;
System.out.println("Dequeued: " + element);
```

```
return element;
}
public boolean isEmpty() {
return size == 0;
}
public boolean isFull() {
return size == capacity;
}
public void display() {
if(isEmpty()) {
System.out.println("Queue is empty.");
return;
}
System.out.print("Queue: ");
for(int i = 0; i < size; i++) {
System.out.print(arr[(front + i) % capacity] + " ");
}
System.out.println();
}
public static void main(String[] args) {
pr11 queue = new pr11(5);
queue.enqueue(150);
queue.enqueue(300);
queue.enqueue(450);
queue.enqueue(600);
queue.display();
queue.dequeue();
queue.display();
queue.enqueue(750);
queue.display();
queue.dequeue();
queue.display();
}
}
```

OUTPUT:

```
Enqueued: 150
Enqueued: 300
Enqueued: 450
Enqueued: 600
Queue: 150 300 450 600
Dequeued: 150
Queue: 300 450 600
Enqueued: 750
Queue: 300 450 600 750
Dequeued: 300
Queue: 450 600 750
```

RESULT:

EX.NO: 12 DATE:	EXCEPTION HANDLING MECHANISM	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
import java.util.*;
public class pr12 {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a number: ");
try {
int num = scanner.nextInt();
int result = 10 / num;
System.out.println("Result = " + result);
} catch (ArithmeticException e) {
System.out.println("Error: " + e.getMessage());
} catch (Exception e) {
System.out.println("Unknown error occurred: " + e.getMessage());
} finally {
System.out.println("Program execution completed.");
}
}
}
```

OUTPUT:

```
Enter a number: 3
Result = 3
Program execution completed.
```

```
Enter a number: 0
Error: / by zero
Program execution completed.
```

RESULT:

EX.NO: 13 DATE:	JSP PROGRAM FOR FIBONACCI SERIES	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

```
<html>
<head><title>FIBONACCI SERIES IN JSP</title></head>
<body>
<form method="get">
<h3> Enter the number of terms you want:
<input type="text" name="limit">
</h3>
</form>
<h3>
<%
 String s = request.getParameter("limit");
 if (s != null) {
%>
<%@ page import = "java.io.*" %>
<%@ p   age import = "java.lang.*" %>
<%
  int n=0;
  n=Integer.parseInt(s);
 out.println("No of terms to be printed is "+n);
%>
<br>
<br>
<br>
The series generated are listed below :<br><br>
<%
  int a=1;
  int b=1;
  out.println(""+a+",\t"+b+",\t");
 for(int i=3;i<= n;i++)
{
  int c=a+b;
  out.print(""+c+",\t");
 a=b;
 b=c;
```

```
}
}
%>
</h3>
</body>
</html>
```

OUTPUT:

No of terms to be printed is 7

The series generated are listed below: 1, 1, 2,

3, 5, 8, 13

RESULT:

EX.NO: 14 DATE:	REQUEST HEADER INFORMATION USING AJAX	PAGE NO:

AIM:

ALGORITHM:

SOURCE CODE:

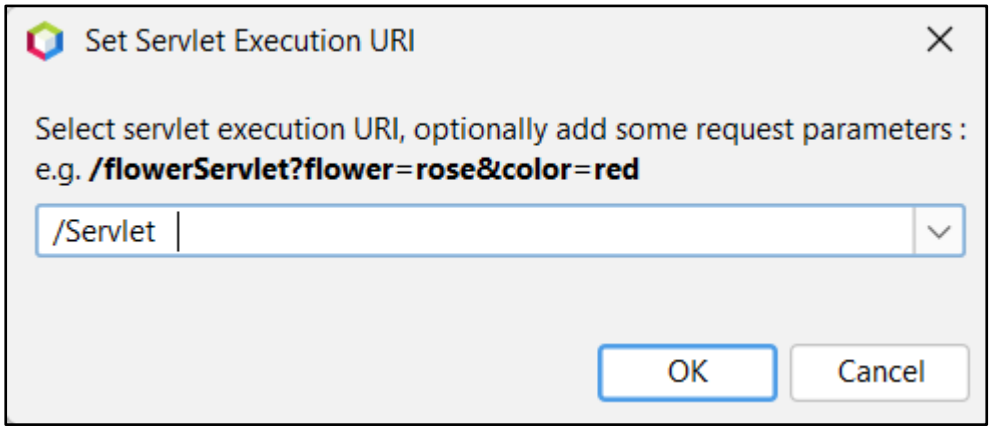
```
import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

* A simple servlet that outputs "Hello World".

* @author DELL

@WebServlet(urlPatterns = {"/Servlet"})

public class NewServlet extends HttpServlet {

/**

* Processes requests for both HTTP <code>GET</code> and <code>POST</code>

* methods.

* @param request servlet request

* @param response servlet response

* @throws ServletException if a servlet-specific error occurs

* @throws IOException if an I/O error occurs

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
```

```
throws ServletException, IOException {
// Set response content type
response.setContentType("text/html;charset=UTF-8");
// Initialize PrintWriter for output
try (PrintWriter out = response.getWriter()) {
// Output "Hello World" to the client
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Hello World Servlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Hello World from Servlet!</h1>");
out.println("</body>");
out.println("</html>");
}
}
//<editor-fold defaultstate="collapsed" desc="HttpServlet methods.
Click on the + sign on the left to edit the code.">
* Handles the HTTP <code>GET</code> method.
```

```
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}
/**
* Handles the HTTP <code>POST</code> method.
* @param request servlet request

* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
```

```
processRequest(request, response);

}

/**

* Returns a short description of the servlet.

*

* @return a String containing servlet description

*/

@Override

public String getServletInfo() {

return "A simple Hello World Servlet";

}// </editor-fold>
```

OUTPUT:



RESULT: