//Practical Number: - 1

//Practical Name :- Write java program that demonstrates generic programming.

---------------------------------------------------------------------------------------------------------------------------------

class MyGen<T>

{

T x,y;

public static<T extends Number>void add(T x,T y)

{

int sum;

sum=x.intValue()+y.intValue();

System.out.println("Addition is "+sum);

}

}

class GenericDemo

{

public static void main(String args[])

{

MyGen<Integer> m=new MyGen<Integer>();

m.add(10,10);

}

}

//Practical No. :- 2.1

//Practical Name :- Write a Java program that demonstrates the use of ArrayList class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

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

obj.add("apple");

obj.add("mango");

obj.add("banana");

Iterator itr=obj.iterator();

while(itr.hasNext())

{

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

}

System.out.println(obj.indexOf("mango"));

obj.remove(0);

System.out.println("after remove list is "+obj);

}

}

//Practical No. :- 2.2

//Practical Name: - Write a Java program that demonstrates the use of LinkedList class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

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

obj.add("apple");

obj.add("mango");

obj.add("banana");

Iterator itr=obj.iterator();

while(itr.hasNext())

{

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

}

System.out.println(obj.indexOf("mango"));

obj.remove(0);

System.out.println("after remove list is "+obj);

}

}

//Practical No. :- 2.3

//Practical Name: - Write a Java program that demonstrates the use of Vector class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

Vector<String> obj=new Vector<String>();

obj.add("apple");

obj.add("mango");

obj.add("banana");

Iterator itr=obj.iterator();

while(itr.hasNext())

{

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

}

System.out.println(obj.indexOf("mango"));

obj.remove(0);

System.out.println("after remove list is "+obj);

}

}

//Practical No:- 2.4

//Practical Name :- Write a Java program that demonstrates the use of Stack class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

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

obj.push("apple");

obj.push("mango");

obj.push("banana");

obj.push("Pineapple");

System.out.println("index of mango is "+obj.indexOf("mango"));

System.out.println("first element is "+obj.firstElement());

Iterator itr=obj.iterator();

while(itr.hasNext())

{

System.out.println(obj.pop());

}

}

}

//Practica No. :- 2.5

//Practical Name :-Write a Java program that demonstrates the use of PriorityQueue class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

PriorityQueue<String> obj=new PriorityQueue<String>();

obj.add("Apple");

obj.add("Mango");

obj.add("Banana");

obj.add("Pineapple");

System.out.println("head "+obj.element());

System.out.println("head "+obj.peek());

System.out.println("Iterating Queue element :- ");

Iterator itr=obj.iterator();

while(itr.hasNext())

{

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

}

obj.remove();

System.out.println("After removing first element Queue is "+obj);

}

}

//Practical No.:- 2.6

//Practical Name:- Write a Java program that demonstrates the use of ArrayDeque class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

ArrayDeque<String> obj=new ArrayDeque<String>();

obj.add("Apple");

obj.add("Mango");

obj.add("Banana");

obj.add("Pineapple");

System.out.println("head "+obj.element());

System.out.println("head "+obj.peek());

System.out.println("Iterating Queue element :- ");

Iterator itr=obj.iterator();

while(itr.hasNext())

{

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

}

obj.remove();

System.out.println("After removing "+obj);

}

}

//Practical No.:- 2.7

//Practical Name:- Write a Java program that demonstrates the use of HashSetclass.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

HashSet<String> obj=new HashSet<String>();

obj.add("Apple");

obj.add("Mango");

obj.add("Banana");

obj.add("Pineapple");

System.out.println("Total element in set is "+obj.size());

System.out.println("Iterating set element :- ");

Iterator itr=obj.iterator();

while(itr.hasNext())

{

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

}

obj.remove("Apple");

System.out.println("After removing "+obj);

}

}

//Practical No.:- 2.8

//Practical Name :- Write a Java program that demonstrates the use of LinkedHashSet class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

LinkedHashSet<String> obj=new LinkedHashSet<String>();

obj.add("Apple");

obj.add("Mango");

obj.add("Banana");

obj.add("Pineapple");

System.out.println("Total element in set is "+obj.size());

System.out.println("Iterating set element :- ");

Iterator itr=obj.iterator();

while(itr.hasNext())

{

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

}

obj.remove("Apple");

System.out.println("After removing "+obj);

}

}

//Practical No.:- 2.9

//Practical Name:- Write a Java program that demonstrates the use of TreeSet class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

TreeSet<String> obj=new TreeSet<String>();

obj.add("Apple");

obj.add("Mango");

obj.add("Banana");

obj.add("Pineapple");

System.out.println("Total element in set is "+obj.size());

System.out.println("Iterating set element :- ");

Iterator itr=obj.iterator();

while(itr.hasNext())

{

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

}

obj.remove("Apple");

System.out.println("After removing "+obj);

}

}

//Practical No.:- 2.10

//Practical Name:- Write a Java program that demonstrates the use of HashMap class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

HashMap<Integer,String> obj=new HashMap<Integer,String>();

obj.put(100,"Apple");

obj.put(101,"Mango");

obj.put(102,"Banana");

obj.put(103,"Pineapple");

System.out.println("Total element in map is "+obj.size());

System.out.println("Elements are as follows");

for(Map.Entry m : obj.entrySet())

{

System.out.println(m.getKey()+"\t"+m.getValue());

}

obj.remove(102);

System.out.println("Updated Map is "+obj);

}

}

//Practical No.:- 2.11

//Practical Name:- Write a Java program that demonstrates the use of LinkedHashMap class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

LinkedHashMap<Integer,String> obj=new LinkedHashMap<Integer,String>();

obj.put(100,"Apple");

obj.put(101,"Mango");

obj.put(102,"Banana");

obj.put(103,"Pineapple");

System.out.println("Total element in map is "+obj.size());

System.out.println("Elements are as follows");

for(Map.Entry m : obj.entrySet())

{

System.out.println(m.getKey()+"\t"+m.getValue());

}

obj.remove(102);

System.out.println("Updated Map is "+obj);

}

}

//Practical no.:- 2.12

//Practical Name:- Write a Java program that demonstrates the use of TreeMap class.

---------------------------------------------------------------------------------------------------------------------------------

import java.util.\*;

public class Demo

{

public static void main(String[] args)

{

TreeMap<Integer,String> obj=new TreeMap<Integer,String>();

obj.put(100,"Apple");

obj.put(101,"Mango");

obj.put(102,"Banana");

obj.put(103,"Pineapple");

System.out.println("Total element in map is "+obj.size());

System.out.println("Elements are as follows");

for(Map.Entry m : obj.entrySet())

{

System.out.println(m.getKey()+"\t"+m.getValue());

}

obj.remove(102);

System.out.println("Updated Map is "+obj);

}

}

//Practical No.:- 3

//Practical Name: - Write a Java program that demonstrates the use of RMI technology.

---------------------------------------------------------------------------------------------------------------------------------

Adder.java

import java.rmi.\*;

public interface Adder extends Remote

{

public int add(int x,int y)throws RemoteException;

}

AdderRemote .java

import java.rmi.\*;

import java.rmi.server.\*;

public class AdderRemote extends UnicastRemoteObject implements Adder

{

AdderRemote()throws RemoteException

{

super();

}

public int add(int x,int y)

{

return x+y;

}

}

MyServer.java

import java.rmi.\*;

import java.rmi.registry.\*;

public class MyServer

{

public static void main(String args[])

{

try

{

Adder stub=new AdderRemote();

Naming.rebind("rmi://localhost:5000/sonoo",stub);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

MyClient.java

import java.rmi.\*;

public class MyClient

{

public static void main(String args[])

{

try

{

Adder stub=(Adder)Naming.lookup("rmi://localhost:5000/sonoo");

System.out.println(stub.add(34,4));

}catch(Exception e){}

}

}

//Practical No. - 4

//Practical Name: - Write a Java program that demonstrates Java Bean.

---------------------------------------------------------------------------------------------------------------------------------

Employee.java

package emp;

public class Employee implements java.io.Serializable

{

private int id;

private String name;

public Employee(){}

public void setId(int id)

{

this.id=id;

}

public int getId()

{

return id;

}

public void setName(String name)

{

this.name=name;

}

public String getName()

{

return name;

}

}

Demo.java

package emp;

public class Demo

{

public static void main(String args[])

{

Employee e=new Employee();//object is created

e.setName("Niraj");//setting value to the object

System.out.println(e.getName());

}

}

//Practical No. - 5

//Practical Name: - Write a Java program(s) that demonstrates EJB.

--------------------------------------------------------------------------------------------------------------------------------

CurrConverterLocal.java

public interface CurrConverterLocal

{

String getRs(String dr);

}

CurrConverter.java

public class CurrConverter implements CurrConverterLocal

{

@Override

public String getRs(String dr)

{

return (Integer.parseInt(dr)\*82)+"";

}

}

Index.html

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<form action="NewServlet" method="get">

Enter Dollar <input type="text" name="d"/>

<br>

<input type="submit">

</form>

</body>

</html>

NewServlet.java

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

String dollar=request.getParameter("d");

out.print(currConverter.getRs(dollar));

}

//Practical No. - 6

//Practical Name: - Write a Java program that demonstrates use of Servlets

---------------------------------------------------------------------------------------------------------------------------------

Index.html

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body bgcolor="yellow">

<form action="NewServlet">

<input type="text" name="fname"/>

<input type="submit" value="click"/>

</form>

</body>

</html>

NewServlet.java

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\*

\* @author admin

\*/

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 {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet NewServlet</title>");

out.println("</head>");

out.println("<body bgcolor=blue>");

out.println("<h1>Welcome " + request.getParameter("fname") + "</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 "Short description";

}// </editor-fold>

}

//Practical No. - 7

//Practical Name: - Write a Java program that demonstrates use of JSP technology.

---------------------------------------------------------------------------------------------------------------------------------

index.html

<html>

<head>

<title>TODO supply a title</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body bgcolor="yellow">

<form action="newjsp.jsp">

Enter First Number <input type="text" name="num1"/>

<br>

Enter Second Number <input type="text" name="num2"/>

<br>

<input type="submit" value="Add"/>

</form>

</body>

</html>

newjsp.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body bgcolor="yellow">

<h1> <%= "Addition is "+(Integer.parseInt(request.getParameter("num1"))+Integer.parseInt(request.getParameter("num2")))%></h1>

</body>

</html>