**PRACTICAL NO:- 1**

**Aim:- Perform the following practicals as mentioned-**

1. **Demonstrate the use of class, object and methods**

**Code:-**

class student{

int rollno;

String name;

void display() {

System.out.println("Student Roll no="+rollno);

System.out.println("Student Name="+name);

}

public static void main(String args[]) {

student a=new student();

a.rollno=72;

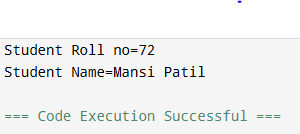
a.name="Mansi Patil";

a.display();

}

}

**Output:-**



1. **Demonstrate the concept of instance of variables.**

**Code:-**

class student{

int rollno;

String name;

void display() {

System.out.println("Student Roll no="+rollno);

System.out.println("Student Name="+name);

}

public static void main(String args[]) {

student a=new student();

student b=new student();

a.rollno=72;

a.name="Mansi Patil";

b.rollno=73;

b.name="Bhawna Patil";

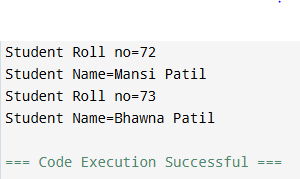
a.display();

b.display();

}

}

**Output:-**



1. **Demonstrate the concept of array**

**Code:-**

class arraydemo {

public static void main(String[] args) {

int number[]={10,20,40,80,160};

for(int i=0;i<number.length;i++)

{

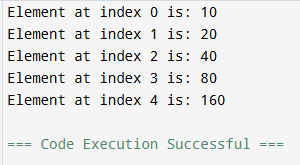
System.out.println("Element at index " +i+ " is: "+number[i]);

}

}

}

**Output:-**



**PRACTICAL NO:- 3**

**AIM:- Perform the following practicals as mentioned:**

**1] Demonstrate method overloading and method overriding in java**

**CODE:-**

class A{

void sum(int a, int b){

System.out.println("sum of" +a+ "&" +b+ "is:"+(a+b));

}

void sum(int x, int y, int z) {

System.out.println("sum of" +x+ "," +y+ "&" +z+ "is:" +(x+y+z));

}

}

class B extends A{

void sum(int a, int b){

System.out.println("sum of" +a+ "&" +b+ "is:"+(a+b));

}

}

public class Main {

public static void main(String[] args){

A a=new A();

B b=new B();

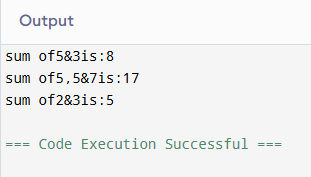
a.sum(5,3);

a.sum(5,5,7);

b.sum(2,3);

}

}



**2] Demonstrate creating your own exception in java**

**A] Throw keyword:**

public class ThrowExample {

static void validateAge(int age) {

if (age < 18) {

throw new IllegalArgumentException("Age must be 18 or older.");

} else {

System.out.println("Age is valid.");

}

}

public static void main(String[] args) {

try {

validateAge(15);

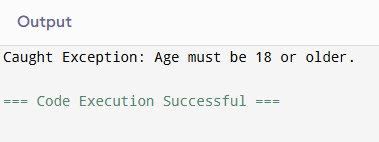
} catch (IllegalArgumentException e) {

System.out.println("Caught Exception: " + e.getMessage());

}

}

}



**B] Throws keyword:**

import java.io.\*;

class Main {

public static void findFile() throws IOException {

File newFile=new File("test.txt");

FileInputStream stream=new FileInputStream(newFile);

}

public static void main(String[] args) {

try{

findFile();

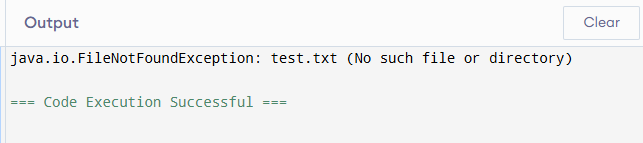
} catch(IOException e){

System.out.println(e);

}

}

}



**C] Try- single catch keyword:-**

public class Main {

public static void main(String[ ] args) {

try {

int[] myNumbers = {1, 2, 3};

System.out.println(myNumbers[10]);

}

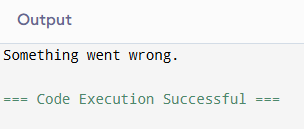
catch (Exception e) {

System.out.println("Something went wrong.");

}

}

}



**D] Try-multiple catch:**

public class Main {

public static void main(String[ ] args) {

try {

int[] myNumbers = {1, 2, 3};

System.out.println(myNumbers[10]);

int result=10/0;

}

catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Caught an ArrayIndexOutOfBoundsException: " + e.getMessage());

}

catch (ArithmeticException e) {

System.out.println("Caught an ArithmeticException: " + e.getMessage());

}

catch (Exception e) {

System.out.println("Something went wrong.");

}

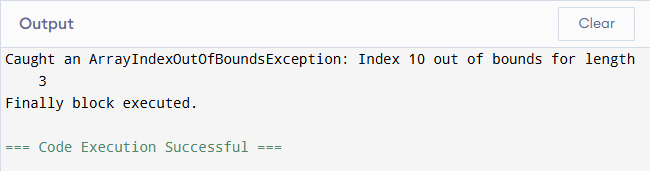
finally {

System.out.println("Finally block executed.");

}

}

}



**PRACTICAl NO:-04**

**AIM:- Perform the following practicals as mentioned:-**

**A] Demonstrate java inheritance using extends keyword**

**Code:**

public class StudentInfo {

void show(int rollno,String name,String div,int mobno){

System.out.println("Roll No: "+rollno);

System.out.println("Name: "+name);

System.out.println("Class: "+div);

System.out.println("Mobile No: "+mobno);

}

}

class cool extends StudentInfo{

public static void main(String []args){

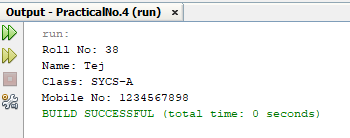
StudentInfo si=new StudentInfo();

si.show(38,"Tej","SYCS-A",1234567898);

}

}

**Output:**



**Practical No-06**

**A] AIM: Write a client socket that will accept n names from user and send them to the server. After receiving the names, the server socket should send the message “names received” and close the connection.**

**CODE:**

**NameServer.java**

package javaapplication11;

import java.io.\*;

import java.net.\*;

public class NameServer {

public static void main(String[] args)throws IOException {

ServerSocket serverSocket=new ServerSocket(5000);

System.out.println("Server started. Wating for clients");

Socket socket=serverSocket.accept();

System.out.println("Client connected.");

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter out=new PrintWriter(socket.getOutputStream(),true);

int n=Integer.parseInt(in.readLine());

System.out.println("Expecting"+n+"names from client...");

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

String name=in.readLine();

System.out.println("Received:"+name);

}

out.println("names received");

socket.close();

serverSocket.close();

System.out.println("Connection closed");

}

}

**NameClient.java**

package javaapplication12;

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class NameClient {

public static void main(String[] args)throws IOException {

Socket socket=new Socket("localhost",5000);

BufferedReader in=new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter out=new PrintWriter(socket.getOutputStream(),true);

Scanner sc=new Scanner(System.in);

System.out.println("Enter number of names: ");

int n=sc.nextInt();

sc.nextLine();

out.println(n);

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

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

String name=sc.nextLine();

out.println(name);

}

String response = in.readLine();

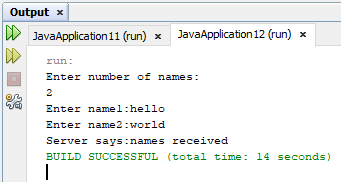
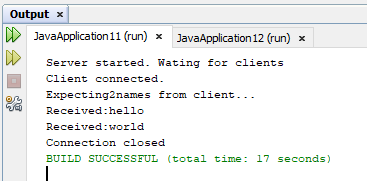
System.out.println("Server says:"+response);

socket.close();

}

}

**OUTPUT:**



**B] AIM: Write a client socket which sends a number to server. The server socket returns the sum of digits of the number if the number is positive, otherwise it sends an error message and close the connection.**

**CODE:**

**DigitServer.java**

import java.io.\*;

import java.net.\*;

public class DigitServer {

public static void main(String[] args) throws IOException {

ServerSocket serverSocket = new ServerSocket(6000);

System.out.println("Digit server started...");

Socket socket = serverSocket.accept();

System.out.println("Client connected.");

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

int number = Integer.parseInt(in.readLine());

if (number >= 0) {

int sum = 0, temp = number;

while (temp > 0) {

sum += temp % 10;

temp /= 10;

}

out.println("Sum of digits = " + sum);

} else {

out.println("Error: Negative number received. Connection closed.");

}

socket.close();

serverSocket.close();

System.out.println("Connection closed.");

}

}

**DigitClient.java**

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class DigitClient {

public static void main(String[] args) throws IOException {

Socket socket = new Socket("localhost", 6000);

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

Scanner sc = new Scanner(System.in);

System.out.print("Enter a number: ");

int number = sc.nextInt();

out.println(number);

String response = in.readLine();

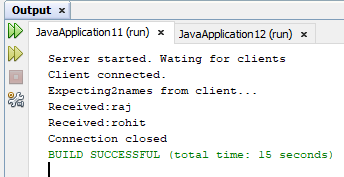
System.out.println("Server says: " + response);

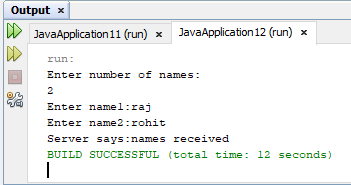
socket.close();

}

}

**OUTPUT:**





**Practical No. 07**

**Aim: Perform the following practical as mentioned:**

**A. Design simple calculator GUI application using AWT components.**

**Code:**

import java.awt.\*;

import java.awt.event.\*;

class Calculator implements ActionListener {

//Declaring Objects

Frame f=new Frame();

Label l1=new Label("First Number");

Label l2=new Label("Second Number");

Label l3=new Label("Result");

TextField t1=new TextField();

TextField t2=new TextField();

TextField t3=new TextField();

Button b1=new Button("Add");

Button b2=new Button("Sub");

Button b3=new Button("Mul");

Button b4=new Button("Div");

Button b5=new Button("Cancel");

Calculator(){

//Giving Coordinates

l1.setBounds(50,100,100,20);

l2.setBounds(50,140,100,20);

l3.setBounds(50,180,100,20);

t1.setBounds(200,100,100,20);

t2.setBounds(200,140,100,20);

t3.setBounds(200,180,100,20);

b1.setBounds(50,250,50,20);

b2.setBounds(110,250,50,20);

b3.setBounds(170,250,50,20);

b4.setBounds(230,250,50,20);

b5.setBounds(290,250,50,20);

//Adding components to the frame

f.add(l1);

f.add(l2);

f.add(l3);

f.add(t1);

f.add(t2);

f.add(t3);

f.add(b1);

f.add(b2);

f.add(b3);

f.add(b4);

f.add(b5);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

f.setLayout(null);

f.setVisible(true);

f.setSize(400,350);

} public void actionPerformed(ActionEvent e) {

int n1=Integer.parseInt(t1.getText());

int n2=Integer.parseInt(t2.getText());

if(e.getSource()==b1){

t3.setText(String.valueOf(n1+n2));

} if(e.getSource()==b2){

t3.setText(String.valueOf(n1-n2));

} if(e.getSource()==b3){

t3.setText(String.valueOf(n1\*n2));

} if(e.getSource()==b4){

t3.setText(String.valueOf(n1/n2));

} if(e.getSource()==b5){

System.exit(0);

}

}

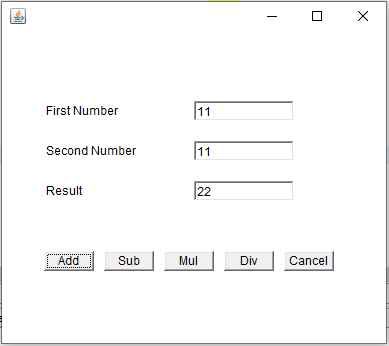
public static void main(String...s) {

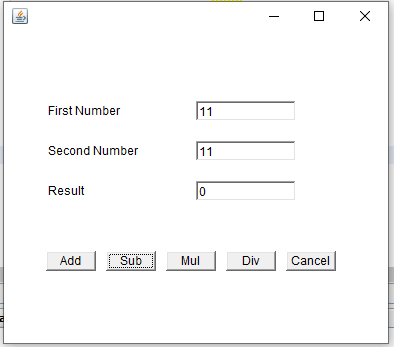
new Calculator();

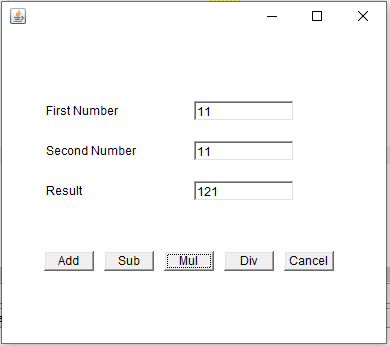
}

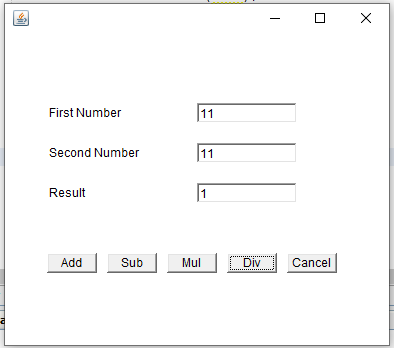
}

**Output:**









**B. Design and implement GUI based application for Pizza order system using AWT components.**

**Code:**

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.BorderFactory;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.JCheckBox;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JRadioButton;

import javax.swing.border.Border;

public class Main extends JFrame {

public static void main(String[] args){

new Main();

}

private JButton buttonOK;

private JRadioButton small, medium, large;

private JCheckBox pepperoni, mushrooms, anchovies;

public Main(){

this.setSize(320,200);

this.setTitle("Order Your Pizza");

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

ButtonListener bl = new ButtonListener();

JPanel mainPanel = new JPanel();

JPanel sizePanel = new JPanel();

Border b1 = BorderFactory.createTitledBorder("Size");

sizePanel.setBorder(b1);

ButtonGroup sizeGroup = new ButtonGroup();

small = new JRadioButton("Small");

small.setSelected(true);

sizePanel.add(small);

sizeGroup.add(small);

medium = new JRadioButton("Medium");

sizePanel.add(medium);

sizeGroup.add(medium);

large = new JRadioButton("Large");

sizePanel.add(large) ;

sizeGroup.add(large);

mainPanel.add(sizePanel);

JPanel topPanel = new JPanel();

Border b2 = BorderFactory.createTitledBorder("Toppings");

topPanel.setBorder(b2);

pepperoni = new JCheckBox("Pepperoni");

topPanel.add(pepperoni);

mushrooms = new JCheckBox("Mushrooms");

topPanel.add(mushrooms);

anchovies = new JCheckBox("Anchovies");

topPanel.add(anchovies);

mainPanel.add(topPanel);

buttonOK = new JButton("OK");

buttonOK.addActionListener(bl);

mainPanel.add(buttonOK);

this.add(mainPanel);

this.setVisible(true);

}

private class ButtonListener implements ActionListener {

public void actionPerformed(ActionEvent e){

if (e.getSource()== buttonOK){

String tops = "";

if (pepperoni.isSelected())

tops +="Pepperoni\n";

if(mushrooms.isSelected())

tops +="Mushrooms\n";

if(anchovies.isSelected())

tops += "Anchovies\n";

String msg = "You ordered a ";

if (small.isSelected())

msg += "small pizza with ";

if (medium.isSelected())

msg += "medium pizza with ";

if (large.isSelected())

msg += "large pizza with ";

if (tops.equals(""))

msg += "no toppings.";

else

msg += "the following toppings:\n" + tops;

JOptionPane.showMessageDialog(buttonOK, msg,"Your Order",JOptionPane.INFORMATION\_MESSAGE);

pepperoni.setSelected(false);

mushrooms.setSelected(false);

anchovies.setSelected(false);

small.setSelected(true);

}

}

}

}

**Output:**

