MAHARISHI DAYANAND UNIVERSITY

Delhi Global Institute of Technology

Java Programming Lab

Submitted By: Bazgha Razi

Subject Code : LC-CSE-327G

Subject Name: Programming in Java Lab

Registration Number :*191380214*

Roll Number :

INDEX

Program 1 **Date: 26/10/2021**

Aim: Create Java program to implement stack and queue concepts.

CODE

Stack Implementation

```
class Stack {
    static final int MAX = 1000;
    int top;
    int a[] = new int[MAX];
    boolean isEmpty()
    {
        return (top < 0);
    }
    Stack()
    {
        top = -1;
    }
    boolean push(int x)
    {
        if (top >= (MAX - 1)) {
            System.out.println("Stack Overflow");
            return false;
        }
        else {
            a[++top] = x;
            System.out.println(x + " pushed into stack");
```

```
        return true;
    }
}
int pop()
{
    if (top < 0) {
        System.out.println("Stack Underflow");
        return 0;
    }
    else {
        int x = a[top--];
        return x;
    }
}
int peek()
{
    if (top < 0) {
        System.out.println("Stack Underflow");
        return 0;
    }
    else {
        int x = a[top];
        return x;
    }
}
```

```
    void print(){
    for(int i = top;i>-1;i--){
      System.out.print(" "+ a[i]); } }      }
class Main {
    public static void main(String args[])
    {
        Stack s = new Stack();
        s.push(10);
        s.push(20);
        s.push(30);
        System.out.println(s.pop() + " Popped from stack");
        System.out.println("Top element is :" + s.peek());
        System.out.print("Elements present in stack :");
        s.print();
    }
}
```

Output

Queue Implementation

Code

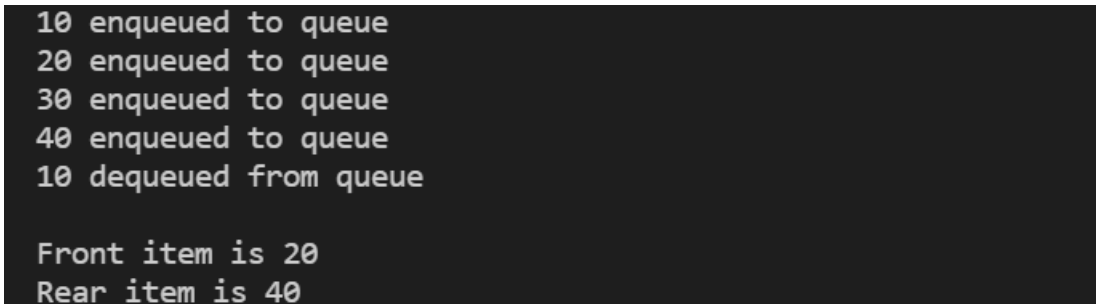
```
class Queue {
```

```
int front, rear, size;
int capacity;
int array[];

public Queue(int capacity)
{
   this.capacity = capacity;
   front = this.size = 0;
   rear = capacity - 1;
   array = new int[this.capacity];
}
boolean isFull(Queue queue)
{
   return (queue.size == queue.capacity);
}
boolean isEmpty(Queue queue)
{
   return (queue.size == 0);
}
void enqueue(int item)
{
   if (isFull(this))
      return;
   this.rear = (this.rear + 1)
            % this.capacity;
   this.array[this.rear] = item;
```

```
        this.size = this.size + 1;
        System.out.println(item+ " enqueued to queue");
    }
    int dequeue()
    {
        if (isEmpty(this))
            return Integer.MIN_VALUE;
        int item = this.array[this.front];
        this.front = (this.front + 1)% this.capacity;
        this.size = this.size - 1;
        return item;
    }
    int front()
    {
        if (isEmpty(this))
            return Integer.MIN_VALUE;
        return this.array[this.front];
    }
    int rear()
    {
        if (isEmpty(this))
            return Integer.MIN_VALUE;
        return this.array[this.rear];
    }
}
public class Test {
```

```
    public static void main(String[] args)
    {
        Queue queue = new Queue(1000);
        queue.enqueue(10);
        queue.enqueue(20);
        queue.enqueue(30);
        queue.enqueue(40);
        System.out.println(queue.dequeue()+ " dequeued from queue\n");
        System.out.println("Front item is "+ queue.front());
        System.out.println("Rear item is "+ queue.rear());
    }
}
```

Output



Program 2 **Date: 02/11/2021**

Aim: Write a java package to show dynamic polymorphism and interfaces.

CODE

Dynamic Polymorphism

```
class Shape
{    void draw()
 {
System.out.println("Drawing Starts"); }  }
 class Rectangle1 extends Shape
 {    void draw()
 {
System.out.println("Now Drawing Rectangle"); }  }
 class Circle1 extends Shape
 {    void draw()
 {
 System.out.println("Now Drawing Circle"); } }
 class Triangle1 extends Shape
 {    void draw()
 {
 System.out.println("Now Drawing Traingle");}
 }    class Shape1
 {    public static void main(String args[])
 { Shape s;
 s=new Rectangle1();
 s.draw();
```

```
s=new Circle1();

s.draw();

s=new Triangle1();

s.draw();  }  }
```

Output

```
Now Drawing Rectangle
Now Drawing Circle
Now Drawing Traingle
```

Interface

Code

```
interface Polygon {
  void getArea(int length, int breadth);
 }
  class Rectangle implements Polygon {
  public void getArea(int length, int breadth) {
   System.out.println("The area of the rectangle is " + (length * breadth));
  }
 }
 class main {
  public static void main(String[] args) {
   Rectangle r1 = new Rectangle();
   r1.getArea(5, 6);
  } }
```

Output

```
The area of the rectangle is 30
```

Program 3 **Date: 02/11/2021**

Aim: Write a java program to show multithreaded producer and consumer application.

CODE

```
public class ProducerConsumerTest
{
 public static void main(String[] args)
 {
 abc c = new abc();
 Producer p1 = new Producer(c, 1);
 Consumer c1 = new Consumer(c, 1);
 System.out.println("*********Program Starts*****************");
 p1.start();
 c1.start(); } }
class abc
{
 private int c;
 private boolean a = false;
 public synchronized int get()
 {
 while (a == false)
 {
 try
 {    wait();
 }
```

```
 catch (Exception e)
 {    e.printStackTrace();     } }
 a = false;
 notifyAll();
 return c;
 }
 public synchronized void put(int value)
 {
 while (a == true)
 {
 try
 {
 wait();
 } catch (Exception e)
 {
 e.printStackTrace();
 }
 }
 c = value;
 a = true;
 notifyAll();
 }
}
class Consumer extends Thread
{
 private abc abc;
```

```
 private int number;
 public Consumer(abc c, int number)
 {
 abc = c;
 this.number = number;
 }
 public void run()
 {
 int value = 0;
 for (int i = 0; i <= 10; i++)
 {
 value = abc.get();
 System.out.println("Consumer got: " + value);
 } } }
class Producer extends Thread
{
 private abc abc;
 private int number;
 public Producer(abc c, int number)
 {
 abc = c;
 this.number = number;
 }
 public void run()
 {
 for (int i = 0; i <= 10; i++)
```

```
{
abc.put(i);
System.out.println("Producer put: " + i);
try
{
sleep((int)(Math.random() * 100));
} catch (InterruptedException e)
{
e.printStackTrace();
}
}
System.out.println("*********Program Ends****************"); } }
```

Output

Program 4 **Date: 09/11/2021**

Aim: Create a customized exception and also make use of all the 5 exception keywords.

CODE

```
import javax.management.InvalidAttributeValueException;

public class ExceptionHandling

{

void trycatch()

{  System.out.println("**********Start of Try and Catch
Block**********\n");

try

{

int a=10;

 int b=0;

 int c=a/b;

 System.out.println("value of c is: "+c);         }

 catch(Exception e)

 {         System.out.println(e+"\n");

 }

finally

{

System.out.println("**********End of Try and Catch
Block**********\n\n\n");   }          }

void validate(int number)throws InvalidAttributeValueException

{ if(number%2!=0)

 throw new InvalidAttributeValueException("The number is not divisble by 2");
```

```
else
 System.out.println("It is divisble by 2");       }
public static void main(String[] args)
{ ExceptionHandling ex= new ExceptionHandling();
ex.trycatch();
System.out.println("**********Start
Throws**********\n");
try  {
 ex.validate(17);
 }
 catch(Exception m)
 {         System.out.println("Exception that occured is: "+m);          }
 System.out.println("Rest of the Code\n");
System.out.println("**********End of
Throws**********");}          }
```

Output

```
**********Start of Try and Catch Block**********

java.lang.ArithmeticException: / by zero

**********End of Try and Catch Block**********



**********Start of Throws**********

Exception that occured is: javax.management.InvalidAttributeValueException: The number is not divisble by 2
Rest of the Code

**********End of Throws**********
```

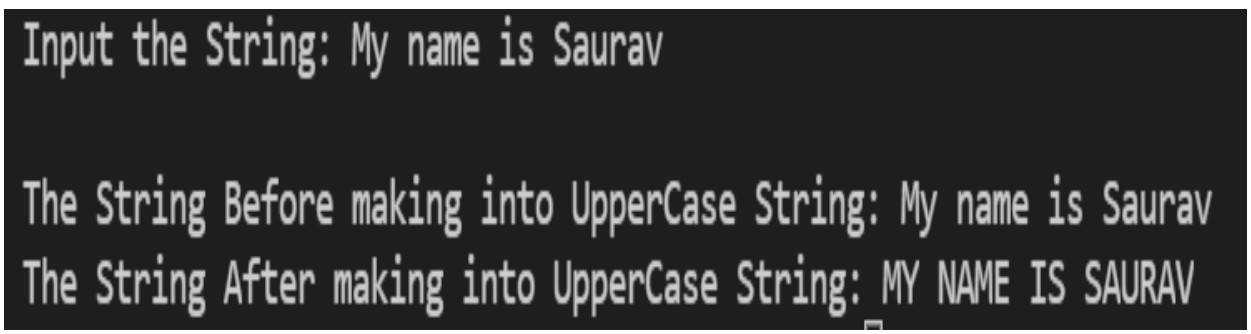
Program 5 **Date: 09/11/2021**

Aim: Convert the content of a given file into the upper case content of the same file.

CODE

```
import java.util.Scanner;

public class StringConv

{

public static void main(String[] args)

{

Scanner s= new Scanner(System.in);

System.out.print("Input the String: ");

String st= s.nextLine();

System.out.println();

System.out.println("The String Before making into UpperCase String: "+st);

st=st.toUpperCase();

System.out.println("The String After making into UpperCase String: "+st);

s.close();

} }
```

Output



Program 6 **Date: 16/11/2021**

Aim: Develop an analog clock using applet.

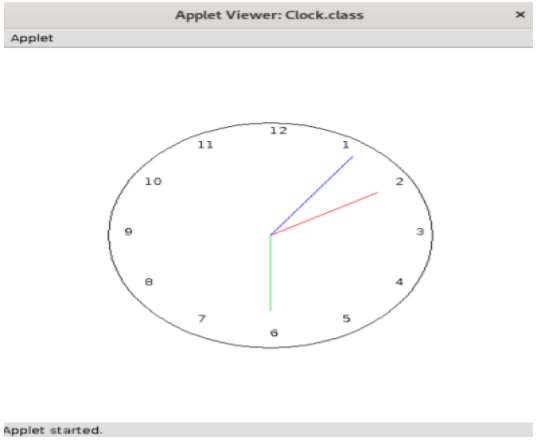
CODE

```
import java.applet.*;
import java.awt.*;
import java.util.*;
public class Clock extends Applet implements Runnable
{
 Thread t;
 public void init()
 {
 setBackground(Color.white);
 }
 public void start()
 {
 t = new Thread(this);
 t.start();
}
 public void run()
{
 while(true)
{
 try
{
 repaint();
```

```
 Thread.sleep(1000);
}
 catch(Exception e)
{
}
}
}
 public void paint(Graphics g)
{
 Calendar time = Calendar.getInstance();
 int hour = time.get(Calendar.HOUR_OF_DAY) % 12;
 int minute = time.get(Calendar.MINUTE);
 int second = time.get(Calendar.SECOND);
 double angle;
 Int x,y;
g.drawOval(100,100,300,300);
 Strings="12";
 inti=0;
 while(i<12)
{
 angle = Math.toRadians(30*(i-3));
x = 250+(int)(Math.cos(angle)*135);
y = 250+(int)(Math.sin(angle)*135);
g.drawString(s,x,y);
i++;
s=String.valueOf(i);}
```

```
g.setColor(Color.green);
 angle = Math.toRadians((30*hour)-90);
x = 250+(int)(Math.cos(angle)*100);
y = 250+(int)(Math.sin(angle)*100);
g.drawLine(250,250,x,y);
g.setColor(Color.red);
 angle = Math.toRadians((6*minute)-90);
x = 250+(int)(Math.cos(angle)*115);
y = 250+(int)(Math.sin(angle)*115);
g.drawLine(250,250,x,y);
g.setColor(Color.blue);
 angle = Math.toRadians((6*second)-90);
x = 250+(int)(Math.cos(angle)*130);
y = 250+(int)(Math.sin
(angle)*130);
g.drawLine(250,250,x,y);
}
}
```

Output



Program 7 **Date: 23/11/2021**

Aim: Develop a scientific calculator using swings.

CODE

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import javax.swing.event.*;
public class ScientificCalculator extends JFrame implements ActionListener
{
JTextField tfield;
double temp, temp1, result, a;
int k = 1, x = 0, y = 0, z = 0;
char ch;
JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, zero, clr, pow2, pow3, exp,
fac, plus, min, div, log, rec, mul, eq, dot, sqrt, sin, cos, tan;
Container cont;
JPanel textPanel, buttonpanel;
ScientificCalculator()
{
cont = getContentPane();
cont.setLayout(new BorderLayout());
JPanel textpanel = new JPanel();
tfield = new JTextField(25);
tfield.setHorizontalAlignment(SwingConstants.RIGHT);
tfield.addKeyListener(new KeyAdapter()
```

```
{
public void keyTyped(KeyEvent keyevent)
{
char
c = keyevent.getKeyChar();
if
(
c >= '0' &&
c <= '9')
{
}
else {
keyevent.consume();
}
}
});
textpanel.add(tfield);
buttonpanel = new JPanel();
buttonpanel.setLayout(new GridLayout(8, 4, 2, 2));
boolean t = true;
b1 = new JButton("1");
buttonpanel.add(b1);
b1.addActionListener(this);
b2 = new JButton("2");
buttonpanel.add(b2);
b2.addActionListener(this);
```

```
b3 = new JButton("3");
buttonpanel.add(b3);
b3.addActionListener(this);
b4 = new JButton("4");
buttonpanel.add(b4);
b4.addActionListener(this);
b5 = new JButton("5");
buttonpanel.add(b5);
b5.addActionListener(this);
b6 = new JButton("6");
buttonpanel.add(b6);
b6.addActionListener(this);
b7 = new JButton("7");
buttonpanel.add(b7);
b7.addActionListener(this);
b8 = new JButton("8");
buttonpanel.add(b8);
b8.addActionListener(this);
b9 = new JButton("9");
buttonpanel.add(b9);
b9.addActionListener(this);
zero = new JButton("0");
buttonpanel.add(zero);
zero.addActionListener(this);
plus = new JButton("+");
buttonpanel.add(plus);
```

```
plus.addActionListener(this);
min = new JButton("-");
buttonpanel.add(min);
min.addActionListener(this);
mul = new JButton("*");
buttonpanel.add(mul);
mul.addActionListener(this);
div = new JButton("/");
div.addActionListener(this);
buttonpanel.add(div);
dot = new JButton(".");
buttonpanel.add(dot);
dot.addActionListener(this);
eq = new JButton("=");
buttonpanel.add(eq);
eq.addActionListener(this);
rec = new JButton("1/x");
buttonpanel.add(rec);
rec.addActionListener(this);
sqrt = new JButton("Sqrt");
buttonpanel.add(sqrt);
sqrt.addActionListener(this);
log = new JButton("log");
buttonpanel.add(log);
log.addActionListener(this);
sin = new JButton("SIN");
```

```
buttonpanel.add(sin);
sin.addActionListener(this);
cos = new JButton
("COS");
buttonpanel.add(cos);
cos.addActionListener(this);
tan = new JButton("TAN");
buttonpanel.add(tan);
tan.addActionListener(this);
pow2 = new JButton("x^2");
buttonpanel.add(pow2);
pow2.addActionListener(this);
pow3 = new JButton("x^3");
buttonpanel.add(pow3);
pow3.addActionListener(this);
exp = new JButton("Exp");
exp.addActionListener(this);
buttonpanel.add(exp);
fac = new JButton("n!");
fac.addActionListener(this);
buttonpanel.add(fac);
clr = new JButton("AC");
buttonpanel.add(clr);
clr.addActionListener(this);
cont.add("Center", buttonpanel);
cont.add("North", textpanel);
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setResizable(false);
}
public void actionPerformed(ActionEvent e)
{
String s = e.getActionCommand();
if(s.equals("1"))
{
if(z == 0){
tfield.setText(tfield.getText() + "1");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "1"); z = 0;
}
}
if(s.equals("2")){
if(z == 0)
{
tfield.setText(tfield.getText() + "2");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "2"); z = 0;
}
}
```

```
if(s.equals("3")){
if(z == 0){
tfield.setText(tfield.getText() + "3");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "3"); z = 0;
}
}
if(s.equals("4")){
if(z == 0){
tfield.setText(tfield.getText() + "4");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "4"); z = 0;
}
}
if
(
s.equals("5"))
{
if
(
z == 0)
{
```

```
tfield.setText(tfield.getText() + "5");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "5"); z = 0;
}
}
if(s.equals("6")) {
if(z == 0){
tfield.setText(tfield.getText() + "6");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "6"); z = 0;
}
}
if(s.equals("7")){
if(z == 0){
tfield.setText(tfield.getText() + "7");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "7"); z = 0;
}
}
if(s.equals("8")) {
```

```
if(z == 0) {
tfield.setText(tfield.getText() + "8");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "8"); z = 0;
}
}
if(s.equals("9")){
if(z == 0){
tfield.setText(tfield.getText() + "9");
}
else {
tfield.setText("");
tfield.setText(tfield.getText() + "9"); z = 0;
}
}
if(s.equals("0")){
if (z == 0){
tfield.setText(tfield.getText() + "0");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "0");
z = 0;
```

```
}
}
if (s.equals("AC"))
{
tfield.setText("");
x = 0;
y = 0;
z = 0;
}
if (s.equals("log")){
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.log(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("1/x")){
if (tfield.getText().equals("")){
tfield.setText("");
}
else
```

```
{
a = 1 / Double.parseDouble(tfield.getText());
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("Exp")) {
if (tfield.getText().equals("")) {
tfield.setText("");
}
else
{
a = Math.exp(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("x^2")) {
if (tfield.getText().equals("")) {
tfield.setText("");
}
else
{
a = Math.pow(Double.parseDouble(tfield.getText()), 2);
tfield.setText("");
tfield.setText(tfield.getText() + a);
```

```
}
}
if (s.equals("x^3")) {
if (tfield.getText().equals("")) {
tfield.setText("");
}
else
{
a = Math.pow(Double.parseDouble(tfield.getText()), 3);
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("+/-"))
{
if (x == 0)
{
tfield.setText("-" + tfield.getText());
x = 1;
}
else
{
tfield.setText(tfield.getText());
}
}
if (s.equals("."))
```

```
{
if (y == 0)
{
tfield.setText(tfield.getText() + ".");
y = 1;
}
else
{
tfield.setText(tfield.getText());
}
}
if (s.equals("+"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
temp = 0;
ch = '+';
}
else
{
temp = Double.parseDouble(tfield.getText());
tfield
.setText("");
ch = '+';
y = 0; x = 0;
```

```
}
tfield.requestFocus();
}
if(s.equals("-")){
if
(tfield.getText().equals(""))
{
tfield.setText("");
temp = 0;
ch = '-';
}
else {
x = 0; y = 0;
temp = Double.parseDouble
(tfield.getText());
tfield.setText("");
ch = '-';
}
tfield.requestFocus();
}
if(s.equals("/")){
if
(tfield.getText().equals(""))
{
tfield.setText("");
temp = 1;
```

```
ch = '/';
}
else {
x = 0;
y = 0;
temp = Double.parseDouble
(tfield.getText());
ch = '/';
tfield.setText("");
}
tfield.requestFocus();
}
if(s.equals("*")){
if
(tfield.getText().equals(""))
{
tfield.setText("");
temp = 1;
ch = '*';
}
else {
x = 0; y = 0;
temp = Double.parseDouble
(tfield.getText());
ch = '*';
tfield.setText("");
```

```
}
tfield.requestFocus();
}
if (s.equals("Sqrt"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.sqrt(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("SIN"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.sin(Double.parseDouble(tfield.getText()));
tfield.setText("");
```

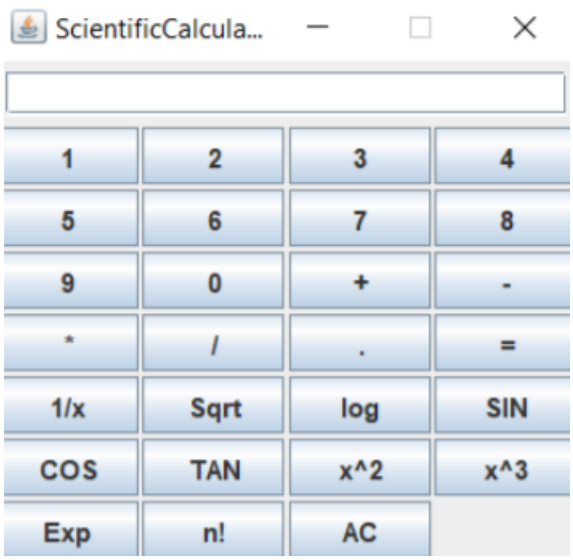
```
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("COS"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.cos(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("TAN"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.tan(Double.parseDouble(tfield.getText()));
tfield.setText("");
```

```
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("="))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
temp1 = Double.parseDouble(tfield.getText());
switch (ch)
{
case '+':
result = temp + temp1;
break;
case '-':
result = temp - temp1;
break;
case '/':
result = temp / temp1;
break;
case '*':
result = temp * temp1;
break;
```

```
}
tfield.setText("");
tfield.setText(tfield.getText() + result);
z = 1;
}
}
if (s.equals("n!"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = fact(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
tfield.requestFocus();
}
double fact(double x)
{
int er = 0;
if (x < 0)
{
```

```
er = 20;

return 0;

}

double i, s = 1;

for (i = 2; i <= x; i += 1.0)

s *= i;

return s;

}

public static void main(String args[])

{

ScientificCalculator f = new ScientificCalculator();

f.setTitle("ScientificCalculatorVarunGupta");

f.pack();

f.setVisible(true);

}

}
```

Output



Program 8 **Date: 30/11/2021**

Aim: Create an editor like MS-word using swings.

CODE

```
import java.awt.*;
import javax.swing.*;
import java.io.*;
import java.awt.event.*;
class editor extends JFrame implements ActionListener
{
JTextArea t;
JFrame f;
editor()
{
f = new JFrame("MS WORD LIKE EDITOR");
t = new JTextArea();
JMenuBar mb = new JMenuBar();
JMenu m1 = new JMenu("File");
JMenuItem m11 = new JMenuItem("New");
JMenuItem m22= new JMenuItem("Open");
JMenuItem m3 = new JMenuItem("Save");
JMenuItem m4 = new JMenuItem("Print");
JMenuItem mc= new JMenuItem("Close");
m11.addActionListener(this);
m22.addActionListener(this);
m3.addActionListener(this);
```

```
m4.addActionListener(this);

mc.addActionListener(this);

m1.add(m11);

m1.add(m22);

m1.add(m3);

m1.add(m4);

m1.add(mc);

JMenu m2 = new JMenu("Edit");

JMenuItem mi4 = new JMenuItem("Cut");

JMenuItem mi5 = new JMenuItem("Copy");

JMenuItem mi6 = new JMenuItem("Paste");

mi4.addActionListener(this);

mi5.addActionListener(this);

mi6.addActionListener(this);

m2.add(mi4);

m2.add(mi5);

m2.add(mi6);

mb.add(m1);

mb.add(m2);

f.setJMenuBar(mb);

f.add(t);

f.setSize(1300, 800);

f.setVisible(true);

f.setResizable(false);

}

public void actionPerformed(ActionEvent e)
```

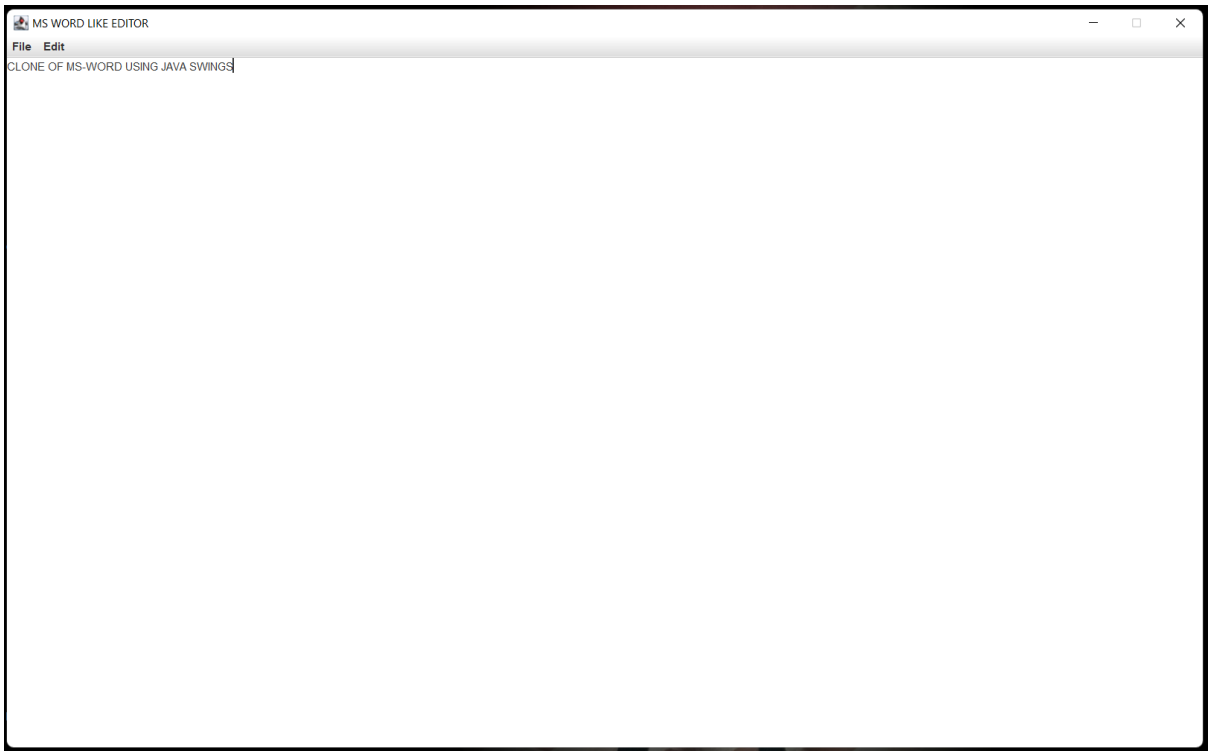
```
{
String s = e.getActionCommand();
if (s.equals("Cut"))
{
t.cut();
}
else if (s.equals("Copy"))
{
t.copy();
}
else if (s.equals("Paste"))
{
t.paste();
}
else if (s.equals("Save"))
{
JFileChooser j = new JFileChooser("f:");
int r = j.showSaveDialog(null);
if (r == JFileChooser.APPROVE_OPTION)
{
File fi = new File(j.getSelectedFile().getAbsolutePath());
try
{
FileWriter wr = new FileWriter(fi, false);
BufferedWriter w = new BufferedWriter(wr);
w.write(t.getText());
```

```
w.flush();
w.close();
}
catch (Exception evt)
{
JOptionPane.showMessageDialog(f, evt.getMessage());
}
}
else
JOptionPane.showMessageDialog(f, "Operation is Cancelled by the User");
}
else if (s.equals("Print"))
{
try
{
t.print();
}
catch (Exception evt)
{
JOptionPane.showMessageDialog(f, evt.getMessage());
}
}
else if (s.equals("Open"))
{
JFileChooser j = new JFileChooser("f:");
int r = j.showOpenDialog(null);
```

```
if (r == JFileChooser.APPROVE_OPTION)
{
File fi = new File(j.getSelectedFile().getAbsolutePath());
try
{
String s1 = "", sl = "";
FileReader fr = new FileReader(fi);
BufferedReader br = new BufferedReader(fr);
sl = br.readLine();
while ((s1 = br.readLine()) != null)
{
sl = sl + "\n" + s1;
}
t.setText(sl);
}
catch (Exception evt)
{
JOptionPane.showMessageDialog(f, evt.getMessage());
}
}
else
JOptionPane.showMessageDialog(f, "Operation is Cancelled by the User");
}
else if (s.equals("New"))
{
t.setText("");
```

```
}

else if (s.equals("Close"))

{

f.setVisible(false);

}

}

public static void main(String args[])

{

editor e = new editor();

}

}
```

Output



Program 9 **Date: 07/12/2021**

Aim: Write a java program to implement arrays and methods.

CODE

```
public class Array

{

public void print(String s[],int a[])

{

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

 {

 System.out.println(s[i]+"->"+a[i]); }   }

 public static void main(String[] args)

 { String[] s =
{"Chips","Apples","Mangoes","Towels","RoomFreshner","Hangers","Pens","CornFlakes","Oa
ts"};

 int[] a= {10,20,40,400,400,150,50,20,25};

 Array ar= new Array();

 ar.print(s,a);

 }

}
```

Output

Program 10 **Date: 21/12/2021**

Aim: Given a word “Javaisbrilliant”. Sort the characters of the given word in alphabetical order and return the sorted word.

CODE

```
import java.util.*;

public class SortingString

{

public static void main(String[] args)

{

String s="Javaisbrilliant";

System.out.print("String before Sorting: "+s);

System.out.println();

char ch[]= s.toCharArray();

Arrays.sort(ch);

System.out.print("String After Sorting: ");

System.out.print(ch);

}

}
```

Output

```
String before Sorting: Javaisbrilliant
String After Sorting: Jaaabiiillnrstv
```

Program 11 **Date: 28/12/2021**

Aim: Write a program which generates following types of exceptions. Demonstrate how these exceptions are handled.
a. ArithmeticException
b. ArrayIndexOutOfBoundsException
c. ArrayStoreException
d. NegativeArraySizeException
e. NullPointerException
f. NumberFormatException.

CODE

```
public class ExceptionQuestion

{

public void exception()

{

try

{

int a=10;

int b=0;

int c=a/b;

System.out.println("c= "+c);

}

catch (ArithmeticException e)

{

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

}

try

{

int ar[]=new int[4];
```

```
ar[5]=0;
System.out.println(ar[5]);
}
catch (ArrayIndexOutOfBoundsException e)
{
System.out.println(e+"\n");
}
try
{
 Number[] a = new Double[2];
 a[0] = 4;
}
catch (ArrayStoreException e)
{
System.out.println(e+"\n");
}
try
{
int ar[]= new int[-4];
System.out.println(ar[2]); }
catch (NegativeArraySizeException e)
{
System.out.println(e+"\n"); }
String a = null;
 try
 {
```

```
if (a.equals("gfg"))
System.out.print("Both are equal");
else
System.out.print("They are not equal");
}
catch(NullPointerException e)
{
System.out.print(e+"\n");
}
try
{
int x= Integer.parseInt("1l");
System.out.println(x);
}
catch (NumberFormatException e)
{
System.out.print(e+"\n"); }    }
public static void main(String[] args){
ExceptionQuestion ex= new ExceptionQuestion();
ex.exception(); }        }
```

Output

```
java.lang.ArithmeticException: / by zero
java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 4
java.lang.ArrayStoreException: java.lang.Integer
java.lang.NegativeArraySizeException: -4
java.lang.NullPointerException
java.lang.NumberFormatException: For input string: "1l"
```

Program 12 **Date: 04/01/2022**

Aim: Create a class to show all the classes of collection framework.

CODE

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Iterator;
import java.util.LinkedHashSet;
import java.util.LinkedList;
import java.util.PriorityQueue;
import java.util.TreeSet;
import java.util.Vector;
public class CollectionQuestions
{
public void arraylist()
{
ArrayList<Integer> a= new ArrayList<Integer>();
 a.add(1);
 a.add(3);
 a.add(5);
 Iterator i= a.iterator();
 System.out.println("ArrayList is: ");
 while(i.hasNext())
 {
 System.out.println(i.next());
 }
```

```
 }
public void vector()
{
Vector<String> a= new Vector<String>();
a.add("Saurav");
a.add("Bamotra");
 Iterator i= a.iterator();
 System.out.println("\nVector is: ");
 while(i.hasNext())
 {
 System.out.println(i.next()); }            }
public void linklist()
{
LinkedList<String> a= new LinkedList<String>();
a.add("ABCD");
a.add("EFGH");
a.add("IJKL");
 Iterator i= a.iterator();
 System.out.println("\nLinkedList is: ");
 while(i.hasNext()) {
 System.out.println(i.next()); }            }
public void priority()
{
PriorityQueue<String> a= new PriorityQueue<String>();
a.add("BTech"); a.add("BBA"); a.add("BJAMC"); a.add("MBA");
 Iterator i = a.iterator();
```

```
 System.out.println("\nPriorityQueue is: ");
 while (i.hasNext()) {
 System.out.println(i.next()); }
a.remove();
 System.out.println("\nPriorityQueue after deleting one element is: ");
 Iterator i1 = a.iterator();
 while (i1.hasNext()){
 System.out.println(i1.next()); }          }
public void hashset() {
HashSet<String>
a = new HashSet<String>();
a.add("Saurav"); a.add("Meenakshi"); a.add("Ashok"); a.add("Gaurav"); a.add("Payal");
Iterator i = a.iterator();
System.out.println("\nHashSet is: ");
while (i.hasNext()){
System.out.println(i.next()); } }
public void linkedhashset() {
LinkedHashSet<String>
a = new LinkedHashSet<String>();
a.add("Saurav"); a.add("Meenakshi"); a.add("Ashok"); a.add("Gaurav"); a.add("Payal");
Iterator i = a.iterator();
System.out.println("\nLinkedHashSet is: ");
while(i.hasNext()){
System.out.println(i.next()); } }
public void treeset() {
TreeSet<String>
```

```
a = new TreeSet<String>();

a.add("Saurav"); a.add("Meenakshi"); a.add("Ashok"); a.add("Gaurav"); a.add("Payal");

Iterator i = a.iterator();

System.out.println("\nTreeSet is: ");

while(i.hasNext()) {

System.out.println(i.next()); }            }

public static void main(String[] args) {

CollectionQuestions c= new CollectionQuestions();

c.arraylist();

c.vector();

c.linklist();

c.priority();

c.hashset();

c.linkedhashset();

c.treeset(); }    }
```

Output

Program 13 **Date: 11/01/2022**

Aim: Use the collection framework to do the following programs:
a. Build and run applications that use "Set" Collection objects.
b. Build and run applications that use "List" Collection objects.
c. Build and run applications that use "Map" Collection objects.

CODE

```
import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.LinkedHashSet;

import java.util.LinkedList;

import java.util.List;

import java.util.Map;

import java.util.Set;

import java.util.TreeSet;

public class SpecificCollection {

List<Integer> list1= new ArrayList<>();

List<String> list2

=new LinkedList<>();

Set<String> set1

=new LinkedHashSet<>();

Set<String> set2

=new TreeSet<>();

Map<Integer, String> map1

=new HashMap<>();

public void arraylist() {
```

```
 list1.add(1);
 list1.add(3);
 list1.add(5);
 list1.add(7);
 Iterator i= list1.iterator();
 System.out.println("ArrayList is: ");
 while(i.hasNext())
{
 System.out.println(i.next());
}
}
public void linklist() {
list2.add("ABCD");
list2.add("EFGH");
list2.add("IJKL");
 Iterator i= list2.iterator();
 System.out.println("\nLinkedList is: ");
 while(i.hasNext())
{
 System.out.println(i.next());
}
}
public void linkedhashset()
{
set1.add("Saurav");
set1.add("Meenakshi");
```

```
set1.add("Ashok");

set1.add("Gaurav");

set1.add("Payal");

Iterator i = set1.iterator();

System.out.println("\nLinkedHashSet is: ");

while(i.hasNext())

{

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

}

}

public void treeset() {

set2.add("Saurav");

set2.add("Meenakshi");

set2.add("Ashok");

set2.add("Gaurav");

set2.add("Payal");

Iterator i = set2.iterator();

System.out.println("\nTreeSet is: ");

while(i.hasNext()) {

System.out.println(i.next()); } }

public void hashmap() {

map1.put(1,"Ashok");

map1.put(2,"Shok");

map1.put(3,"Hok");

map1.put(4,"Ok");

map1.put(5, "K");
```

```
System.out.println("\nHashmap is:");

for(Map.Entry m : map1.entrySet()) {

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

public static void main(String[] args) {

SpecificCollection sc= new SpecificCollection();

sc.arraylist();

sc.linklist();

sc.linkedhashset();

sc.treeset();

sc.hashmap(); }          }
```

Output

```
ArrayList is:
1
3
5
7

LinkedList is:
ABCD
EFGH
IJKL

LinkedHashSet is:
Saurav
Meenakshi
Ashok
Gaurav
Payal

TreeSet is:
Ashok
Gaurav
Meenakshi
Payal
Saurav

Hashmap is:
1 Ashok
2 Shok
3 Hok
4 Ok
5 K
```

Program 14 **Date: 11/01/2022**

Aim: Write a java program for implementation of JDBC.

CODE

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Scanner;
import java.sql.PreparedStatement;
interface JDBCD
{
void update();
void delete();
}
public class JDBC implements JDBCD
{
int id1,nos,id2,nos1;
String cn,pd,certno,fh,sh;
Double pr,cr;
Scanner s = new Scanner(System.in);
Scanner s1 = new Scanner(System.in);
Scanner s2 = new Scanner(System.in);
Scanner s3 = new Scanner(System.in);
public void master()
{
```

```
System.out.print("Enter the ShareID: ");

id1=s.nextInt();

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

cn=s1.nextLine();

System.out.print("\nEnter the No Of Shares: ");

nos=s.nextInt();

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

pd=s2.nextLine();

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

pr= s.nextDouble();

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

cr=s.nextDouble();

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection co=
DriverManager.getConnection("jdbc:mysql://localhost:3306/practical","root", "root");

Statement st=co.createStatement();

st.executeUpdate("create table if not exists master(ShareID int(3) Unique,"+ "CompName
varchar(20)," +"NoOFShares int(4),"+"PurRatedouble(6,2)," + "PurDate date," + "CurrRate
double(6,2))");

st.executeUpdate("insert into
mastervalues('"+id1+"','"+cn+"','"+nos+"','"+pr+"','"+pd+"','"+cr+"')");

System.out.println("Master Table Details Inserted Successfully!!!!!\n\n");

}

catch (Exception e1)

{
```

```
System.out.println("error is "+e1);
e1.printStackTrace();
}
}
public void Details()
{
System.out.print("Enter the ShareID: ");
id2=s.nextInt();
System.out.print("\nEnter the Certificate Number: ");
certno=s1.nextLine();
System.out.print("\nEnter the No Of Shares: ");
nos1=s.nextInt();
System.out.print("\nEnter the First Holder Name: ");
fh=s2.nextLine();
System.out.print("\nEnter the Second Holder Name: ");
sh= s3.nextLine();
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection co=
DriverManager.getConnection("jdbc:mysql://localhost:3306/practical","root", "root");
Statement st=co.createStatement();
st.executeUpdate("create table if not exists details(ShareID int(3) Unique,"+ "CertNo
varchar(20)," +"NoOFShares int(4),"+"FirstHoldervarchar(15)," + "SecondHolder
varchar(15))");
st.executeUpdate("insert into
detailsvalues('"+id2+"','"+certno+"','"+nos1+"','"+fh+"','"+sh+"')");
```

```
System.out.println("Details Table Information Inserted Successfully!!!!!\n\n");
}
catch (Exception e1)
{
System.out.println("error is "+e1);
e1.printStackTrace();
}
}
public void alter()
{
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection co=
DriverManager.getConnection("jdbc:mysql://localhost:3306/practical","root", "root");
Statement st=co.createStatement();
String delete="Alter table details drop column CertNo";
String insert="Alter table details add CertNo Varchar(13)";
st.executeUpdate(delete);
st.executeUpdate(insert);
System.out.println("Details Table Datatype Updated Successfully!!!!!\n\n");
}
catch (Exception e1)
{
System.out.println("error is "+e1);
e1.printStackTrace();
```

```
}

}

public void display()

{

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection co=
DriverManager.getConnection("jdbc:mysql://localhost:3306/practical","root", "root");

Statement st=co.createStatement();

ResultSet rs= st.executeQuery("select * from master,details where
master.ShareID=details.ShareID");

while(rs.next())

{

System.out.println();

int ID=rs.getInt("ShareID");

String CN=rs.getString("CompName");

int NOS=rs.getInt("NoOFShares");

String PD=rs.getString("PurDate");

Double PR=rs.getDouble("PurRate");

Double CR=rs.getDouble("CurrRate");

String FH=rs.getString("FirstHolder");

String SH=rs.getString("SecondHolder");

String CertNo=rs.getString("CertNo");

System.out.println("ShareID: "+ID);

System.out.println("Company Name: "+CN);

System.out.println("Number Of Shares: "+NOS);
```

```
System.out.println("Purchased Date: "+PD);
System.out.println("Purchased Rate: "+PR);
System.out.println("Current Rate: "+CR);
System.out.println("Certificate Number: "+CertNo);
System.out.println("First Holder: "+FH);
System.out.println("Second Holder: "+SH);
}
}
catch (Exception e1)
{
System.out.println("error is "+e1);
e1.printStackTrace();
}
}
public void insert()
{
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection co= DriverManager.getConnection("jdbc:mysql://localhost:3306/practical",
"root", "root");
Statement st=co.createStatement();
Scanner s5= new Scanner(System.in);
System.out.println("\nEnter the Insert Statement: ");
String insert=s5.nextLine();
st.executeUpdate(insert);
```

```
System.out.println("Insertion Successfully!!!!");

s5.close();

}

catch (Exception e1)

{

System.out.println("error is "+e1);

e1.printStackTrace();

}

}

public void update()

{

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection co=
DriverManager.getConnection("jdbc:mysql://localhost:3306/practical","root", "root");

System.out.println("\nUpdating Name of the Company");

PreparedStatement ps= co.prepareStatement("update master set CompName=?
whereShareID=?");

ps.setString(1, "Google");

ps.setInt(2, 2);

ps.execute();

System.out.println("Updation Successfully Done!!!!!!");

}

catch (Exception e1)

{

System.out.println("error is "+e1);
```

```
e1.printStackTrace();
}
}
public void delete()
{
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection co= DriverManager.getConnection("jdbc:mysql://localhost:3306/practical",
"root", "root");
PreparedStatement ps=co.prepareStatement("delete from master where ShareID=?");
System.out.println("Deleting a row from master table");
ps.setInt(1, 2);
ps.execute();
System.out.println("Deletion Successfully Done!!!!!");
}
catch (Exception e1)
{
System.out.println("error is "+e1);
e1.printStackTrace();
}
}
public static void main(String[] args)
{
JDBC j= new JDBC();
Scanner s4 = new Scanner(System.in);
```

```
System.out.println("Contents");
System.out.println("1. Insert Details into Tables");
System.out.println("2. Change Datatype of Certificate to varchar(13)");
System.out.println("3. Insert statements can be given at runtime");
System.out.println("4, Display all the Rows of Master Table");
System.out.println("5. Design a program interface to Delete,Update,Insert from interface");
System.out.print("Enter your Choice: ");
int choice=s4.nextInt();
if(choice==1)
{
j.master();
j.Details();
}
if(choice==2)
{
j.alter();
}
if(choice==3)
{
j.insert();
}
if(choice==4)
{
j.display();
}
if(choice==5)
```

```
{
System.out.println("1. Update");
System.out.println("2. Delete");
Scanner s=new Scanner(System.in);
System.out.print("Enter your Choice: ");
int a= s.nextInt();
if(a==1)
{
j.update();
}
else if(a==2)
{
j.delete();
}
s.close();
}
s4.close();
}
}
```

Output

```
Contents
1. Insert Details into Tables
2. Change Datatype of Certificate to varchar(13)
3. Insert statements can be given at runtime
4, Display all the Rows of Master Table
5. Design a program interface to Delete,Update,Insert from interface
Enter your Choice: 2
```

```
mysql> desc details;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| ShareID       | int(3)       | YES  | UNI | NULL    |       |
| NoOFShares    | int(4)       | YES  |     | NULL    |       |
| FirstHolder   | varchar(15)  | YES  |     | NULL    |       |
| SecondHolder  | varchar(15)  | YES  |     | NULL    |       |
| CertNo        | varchar(13)  | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
5 rows in set (0.13 sec)
```

```
Contents
1. Insert Details into Tables
2. Change Datatype of Certificate to varchar(13)
3. Insert statements can be given at runtime
4, Display all the Rows of Master Table
5. Design a program interface to Delete,Update,Insert from interface
Enter your Choice: 5
1. Update
2. Delete
Enter your Choice: 1
```

```
mysql> select * from master;
+---------+-----------+------------+---------+------------+----------+
| ShareID | CompName  | NoOFShares | PurRate | PurDate    | CurrRate |
+---------+-----------+------------+---------+------------+----------+
|       1 | Tesla     |        543 |  234.68 | 2020-09-09 |   123.65 |
|       2 | Microsoft |        123 |  876.56 | 2021-01-19 |   924.84 |
|       3 | Jaguar    |        345 |  654.76 | 2020-07-23 |   978.54 |
+---------+-----------+------------+---------+------------+----------+
3 rows in set (0.00 sec)

mysql> select * from master;
+---------+----------+------------+---------+------------+----------+
| ShareID | CompName | NoOFShares | PurRate | PurDate    | CurrRate |
+---------+----------+------------+---------+------------+----------+
|       1 | Tesla    |        543 |  234.68 | 2020-09-09 |   123.65 |
|       2 | Google   |        123 |  876.56 | 2021-01-19 |   924.84 |
|       3 | Jaguar   |        345 |  654.76 | 2020-07-23 |   978.54 |
+---------+----------+------------+---------+------------+----------+
3 rows in set (0.00 sec)
```