**JAVA PRACTICALS**

**Practical - 1:**

**Aim: Static, Function overloading, Constructor overloading, function overriding**

**Creating & calling (invoking) static method**

public class StaticTest {

  // class method

  static void myStaticMethod() {

    System.out.println("Static methods can be called without creating objects");

  }

  // instance method

  public void myPublicMethod() {

    System.out.println("Public methods must be called by creating objects");

  }

  public static void main(String[ ] args) {

       myStaticMethod(); // Call the static method

    // myPublicMethod(); This would output an error

    StaticTest obj = new StaticTest (); // Create an object of Main

    obj.myPublicMethod(); // Call the public method

  }

}

**Design a class to overload a method volume( ) as follows:**

• double volume(double r) — with radius (r) as an argument, returns the volume of sphere using the formula:  
v = (4/3) \* (22/7) \* r \* r \* r

• double volume(double h, double r) — with height(h) and radius(r) as the arguments, returns the volume of a cylinder using the formula:  
v = (22/7) \* r \* r \* h

• double volume(double 1, double b, double h) — with length(l), breadth(b) and height(h) as the arguments, returns the volume of a cuboid using the formula:  
v = l\*b\*h ⇒ (length \* breadth \* height)

public class Volume {

double v;

    double volume(double r) {

         v = (4 / 3.0) \* (22 / 7.0) \* r \* r \* r;

        return v;

    }

    double volume(double h, double r) {

         v = (22 / 7.0) \* r \* r \* h;

        return v;

    }

      double volume(double l, double b, double h) {

        v = l \* b \* h;

        return v;

      }

    public static void main(String args[]) {

        Volume obj = new Volume();

        System.out.println("Sphere Volume = " + obj.volume(6));

        System.out.println("Cylinder Volume = " + obj.volume(5, 3.5));

        System.out.println("Cuboid Volume = " +  obj.volume(7.5, 3.5, 2));

    }

}

**Constructor overloading**

public class Main {

  String language;

  // constructor with no parameter

  Main() {

    this.language = "Java";

  }

  // constructor with a single parameter

  Main(String language) {

    this.language = language;

  }

  public void getName() {

    System.out.println("Programming Langauage: " + this.language);

  }

  public static void main(String[] args) {

    // call constructor with no parameter

    Main obj1 = new Main();

    // call constructor with a single parameter

    Main obj2 = new Main("Python");

    obj1.getName();

    obj2.getName();

  }

}

**Inheritance & method overriding**

class Animal {

  public void animalSound() {

    System.out.println("The animal makes a sound");

  }

}

class Pig extends Animal {

  public void animalSound() {

    System.out.println("The pig says: wee wee");

  }

}

class Dog extends Animal {

  public void animalSound() {

    System.out.println("The dog says: bow wow");

  }

}

public class OverloadTest {

  public static void main(String[] args) {

Pig obj1 = new Pig();

obj1.animalSound();

Dog obj2 = new Dog();

obj2.animalSound();

  }

}

**Practical - 2:**

**Aim: Write the four programs. in your notebook & bring completed notebook on practical day.**

**Abstract class & abstract method**

abstract class GraphicObject {

   int x, y;

    void moveTo(int newX, int newY) {

      x = newX;

      y = newY;

    }

    abstract void draw();

    abstract void resize();

}

class Circle extends GraphicObject {

    //overriding draw & reshape method

    public  void draw() {

         System.out.println(“draw circle”);

    }

    public  void resize() {

           System.out.println(“resize circle”);

    }

}

class Rectangle extends GraphicObject {

    //overriding draw & reshape method

    public  void draw() {

System.out.println(“draw rect”);

    }

    public  void resize() {

System.out.println(“draw rect”);

    }

}

class AbstracDemo {

    public static void main(String[] args) {

        Circle obj1 = new Circle ();

obj1.draw();

obj1.resize();

        Rectangle  obj2 = new Rectangle  ();

obj2.draw();

obj2.resize();

    }

}

================

**Interface**

interface Animal {

  //public static final field

  int VAL = 20;

  //public abstract  methods

  void animalSound(); // interface method (does not have a body)

  void sleep();

}

// Pig "implements" the Animal interface

class Pig implements Animal {

  //overriding animalSound & sleep method

  public void animalSound() {

    System.out.println("The pig says: wee wee");

  }

  public void sleep() {

    System.out.println("Zzz");

  }

}

public class InterfaceTest {

  public static void main(String[] args) {

    Pig myPig = new Pig();  // Create a Pig object

    myPig.animalSound();

    myPig.sleep();

  }

}

=======================

**Inheriting two interfaces**

interface FirstInterface {

  public void myMethod(); // interface method

}

interface SecondInterface {

  public void myOtherMethod(); // interface method

}

class DemoClass implements FirstInterface, SecondInterface {

//must give body to myMethod, myOtherMethod method

  public void myMethod() {

    System.out.println("Some text..");

  }

  public void myOtherMethod() {

    System.out.println("Some other text...");

  }

}

class Interface2Test {

  public static void main(String[] args) {

    DemoClass myObj = new DemoClass();

    myObj.myMethod();

    myObj.myOtherMethod();

  }

}

===============

**User defined Exception**

class MyException extends Exception {

private int detail;

MyException(int a) {

detail = a;

}

public String toString() {

return "MyException[" + detail + "]";

}

}

class ExceptionDemo {

static void compute(int a) throws MyException {

System.out.println("Called compute(" + a + ")");

if(a > 10)

throw new MyException(a);

System.out.println("Normal exit");

}

public static void main(String args[]) {

try {

compute(1);

compute(20);

} catch (MyException e) {

System.out.println("Caught " + e);

}

finally {

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

}

}

**Practical – 3:**

**Aim: Write the four programs. in your Sheet & bring completed one on practical day.**  
**COLLECTION FRAMEWORK**

**ArrayList**

import java.util.\*;

class ArrayListDemo {

    public static void main(String args[]) {

// Create an array list.

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

        System.out.println("Initial size of al: "+ al.size());

// Add elements to the array list.

        al.add("C");

        al.add("A");

        al.add("E");

        al.add("B");

        al.add("D");

        al.add("F");

        al.add(1, "A2");

        System.out.println("Size of al after additions: " + al.size());

// Display the array list.

        System.out.println("Contents of al: " + al);

// Remove elements from the array list.

        al.remove("F");

        al.remove(2);a

        System.out.println("Size of al after deletions: "+ al.size());

        System.out.println("Contents of al: " + al);

    }

}

**Treeset**

import java.util.\*;

class HashSetDemo {

public static void main(String args[]) {

// Create a hash set.

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

// Add elements to the hash set.

hs.add("B"); hs.add("Af"); hs.add("Aa"); hs.add("D"); hs.add("C"); hs.add("F");

System.out.println(hs);

    for(String s: hs) {

        System.out.println(s);

}

}

}

**Storing User-Defined Classes in Collections**

import java.util.\*;

class Address {

private String name;

private String street;

private String city;

private String state;

private String code;

Address(String n, String s, String c,

String st, String cd) {

name = n;

street = s;

city = c;

state = st;

code = cd;

}

public String toString() {

return name + "\n" + street + "\n" + city + " " + state + " " + code;

}

}

class MailList {

          public static void main(String args[]) {

LinkedList<Address> ml = new LinkedList<Address>();

// Add elements to the linked list.

ml.add(new Address("J.W. West", "11 Oak Ave", "Urbana", "IL", "61801"));

ml.add(new Address("Ralph Baker", "1142 Maple Lane", "Mahomet", "IL", "61853"));

ml.add(new Address("Tom Carlton", "867 Elm St", "Champaign", "IL", "61820"));

// Display the mailing list.

for(Address element : ml) {

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

System.out.println();   }

}

}

**HashMap**

import java.util.\*;

class HashMapDemo {

public static void main(String args[]) {

// Create a hash map.

HashMap<String, Double> hm = new HashMap<String, Double>();

// Put elements to the map

hm.put("John Doe", new Double(3434.34));

hm.put("Tom Smith", new Double(123.22));

hm.put("Jane Baker", new Double(1378.00));

hm.put("Tod Hall", new Double(99.22));

hm.put("Ralph Smith", new Double(-19.08));

// Get a set of the entries.

Set<Map.Entry<String, Double>> set = hm.entrySet();

// Display the set.

for(Map.Entry<String, Double> me : set) {

System.out.print(me.getKey() + ": ");

System.out.println(me.getValue());

}

System.out.println();

// Deposit 1000 into John Doe's account.

double balance = hm.get("John Doe");

hm.put("John Doe", balance + 1000);

System.out.println("John Doe's new balance: " + hm.get("John Doe"));

}

}

**Practical – 4:**

**Aim: Swing**

**Color Changer  JFrame with three JComboBox.**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class ColorDemo implements ItemListener {

    JComboBox<Integer> r,g,b;

    JFrame jfrm ;

    ColorDemo() {

        jfrm = new JFrame("Color changer");

        jfrm.setSize(275, 100);

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

        jfrm.setLayout(new FlowLayout());

        r = new JComboBox<>();

        g = new JComboBox<>();

        b = new JComboBox<>();

        jfrm.add(r);

        jfrm.add(g);

        jfrm.add(b);

        for (int i = 0; i < 256; i++) {

            r.addItem(i);

            g.addItem(i);

            b.addItem(i);

        }

        r.addItemListener(this);

        g.addItemListener(this);

        b.addItemListener(this);

        // Display the frame.

         jfrm.setVisible(true);

    }

    public static void main(String args[]) {

       new ColorDemo();

    }

    @Override

    public void itemStateChanged(ItemEvent e) {

        int rc= r.getSelectedIndex();

        int gc= g.getSelectedIndex();

        int bc= b.getSelectedIndex();

        System.out.println(rc +" "+ gc);

        Color c = new Color (rc,gc,bc);

        jfrm.getContentPane().setBackground(c);

    }

}

**Calculator**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Calc extends JFrame implements ActionListener

{

    public static void main(String args[]) {

        new Calc();

    }

    JButton add,sub,pro,div,clr;

    JTextField t1,t2;

    JLabel res;

    public Calc() {

        setTitle(" Calculator ");

        add=new JButton("    +    ");

        sub=new JButton("    -    ");

        pro=new JButton("    \*    ");

        div=new JButton("    /    ");

        clr=new JButton("  clear  ");

        t1=new JTextField(20);

        t2=new JTextField(20);

        res=new JLabel();

        JPanel top=new JPanel();

        top.setBackground(Color.yellow);

        JPanel bot=new JPanel();

        bot.setBackground(Color.blue);

        bot.setLayout(new FlowLayout());

        bot.add(add);

        bot.add(sub);

        bot.add(pro);

        bot.add(div);

        bot.add(clr);

        bot.add(res);

        top.setLayout(new GridLayout(2,2));

        top.add(new JLabel("Number1:"));

        top.add(t1);

        top.add(new JLabel("Number2:"));

        top.add(t2);

        setLayout(new GridLayout(2,1));

        add(top);

        add(bot);

        setSize(400,150);

        setVisible(true);

        add.addActionListener(this);

        sub.addActionListener(this);

        pro.addActionListener(this);

        div.addActionListener(this);

        clr.addActionListener(this);

        setDefaultCloseOperation(EXIT\_ON\_CLOSE);

    }

    public void actionPerformed(ActionEvent e) {

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

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

        int r=0;

        if(e.getSource()==add){

        r=n1+n2;

        res.setText(r+"");

        }

        else if(e.getSource()==sub){

        r=n1-n2;

        res.setText(r+"");

        }

        else if(e.getSource()==pro){

        r=n1\*n2;

        res.setText(r+"");

        }

        else if(e.getSource()==div){

        r=n1/n2;

        res.setText(r+"");

        }

        else{

            t1.setText("");

            t2.setText("");

            res.setText("");

        }

    }

}

**Resume Builder**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Resume extends JFrame implements ActionListener{

     public static void main(String args[]) {

        new Resume();

    }

    JButton sub,clr;

    JTextField nm,bd,fn,nas,ph,em;

    JRadioButton m, f;

    JComboBox<String> q;

    public Resume() {

        setTitle("Resume builder");

        setDefaultCloseOperation(EXIT\_ON\_CLOSE);

        setLayout(new GridLayout(10,2));

        add(new JLabel("Student's Name"));

        nm = new JTextField(25);

        add(nm);

        add(new JLabel("Father's Name"));

        fn = new JTextField(25);

        add(fn);

        add(new JLabel("Birthday"));

        bd = new JTextField(10);

        add(bd);

        add(new JLabel("Nationality"));

        nas = new JTextField(25);

        add(nas);

        add(new JLabel("Phone "));

        ph = new JTextField(25);

        add(ph);

        add(new JLabel("E-Mail "));

        em = new JTextField(10);

        add(em);

        m = new JRadioButton("Male");

        f = new JRadioButton("Female");

        ButtonGroup bg = new ButtonGroup();

        //add(new JLabel("Gender "));

        bg.add(m);

        bg.add(f);

        add(m);

        add(f);

        add(new JLabel("Qualification"));

        q= new JComboBox<>();

        q.addItem("A");

        q.addItem("B");

        q.addItem("C");

        add(q);

        sub = new JButton("Submit");

        sub.addActionListener(this);

        add(sub);

        clr = new JButton("Clear");

        clr.addActionListener(this);

        add(clr);

        ta = new JTextArea();

        JScrollPane sp=new JScrollPane(ta, JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED,

                JScrollPane.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED);

        add(sp);

        pack();

        setVisible(true);

    }

JTextArea ta;

    @Override

    public void actionPerformed(ActionEvent e) {

        if (e.getSource()== sub) {

            StringBuffer b = new StringBuffer();

            b.append(nm.getText() + '\n');

            b.append(fn.getText() + '\n');

            b.append(bd.getText() + '\n');

            b.append(nas.getText() + '\n');

            b.append(ph.getText() + '\n');

            b.append(em.getText() + '\n');

            b.append(bd.getText() + '\n');

            b.append(q.getSelectedItem() + "\n");

            ta.setText(b.toString());

        }

        else {

            nm.setText("");

            fn.setText("");

            bd.setText("");

            nas.setText("");

            ph.setText("");

            em.setText("");

            bd.setText("");

        }

    }

}

**Practical – 5:**

**Aim: JDBC**

**Create DataBase ‘mydb1’ in MySql**

**Create table ‘emp’ with following fields & records.**

**empid**

**ename**

**dept**

**salary**

1

Abc

Sales

100000

2

Xyz

IT

100000

CREATE DATABASE mydb1;

use mydb1;

CREATE TABLE emp (

    empid int,

    ename varchar(255),

    dept varchar(255),

    salary varchar(255)

);

INSERT INTO emp (empid, ename, dept, salary)

values (1, ‘abc’, ‘sales’, ‘100000’);

INSERT INTO emp (empid, ename, dept, salary)

values (2, ‘xyz’, ‘IT’, ‘100000’);

**//prints all record of emp table present in mydb1 DB using executeQuery()**

import java.sql.\*;

class MysqlCon{

public static void main(String args[]){

try{

    try{

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

    }

    }catch(ClassNotFoundException ce) { System.out.println(ce);}

Connection con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/mydb1","root","cs");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from emp");

while(rs.next())

System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+

rs.getString(3) + rs.getInt(4);

con.close();

}catch(SqlException se) { System.out.println(se);}

}

}

**// inserting a record in emp table present in mydb1 DB using executeUpdate ()**

import java.sql.\*;

class FetchRecord{

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

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

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb1 ","root","cs");

Statement stmt=con.createStatement();

int result=stmt.executeUpdate("insert into emp values(3,'Irfan', ‘IT’ ,200000)");

System.out.println(result+" records affected");

con.close();

} }

**// inserting values by user in emp table by using prepareStatement()**

import java.sql.\*;

class InsertPrepared{

public static void main(String args[]){

try{

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

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb1 ","root","cs");

PreparedStatement stmt=con.prepareStatement("insert into emp values(?,?)");

stmt.setInt(1,101);//1 specifies the first parameter in the query

stmt.setString(2,"Ratan");

int i=stmt.executeUpdate();

System.out.println(i+" records inserted");

con.close();

}catch(Exception e){ System.out.println(e);}

}

}

**// GUI program to add record in table by using PreparedStatement**

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import java.sql.\*;

public class RecordAdd extends JFrame {

    private JTextField name, eid, dept, salary;

    private JButton add, exit;

    public RecordAdd() {

       addC();

        add.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                int id = Integer.parseInt(eid.getText());

     String n = name.getText();

     String d = dept.getText();

                int s = Integer.parseInt(salary.getText());

                try {

                    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb1 ","root","cs");

                    PreparedStatement ps = con.prepareStatement(

                    "INSERT INTO emp (empid, ename, dept, salary) VALUES (?, ?, ?, ?)");

                    ps.setInt(1, id);

                    ps.setString(2, n);

         ps.setString(3, d);

         ps.setInt(4, s);

                    int i= ps.executeUpdate();

if(i > 0)

 JOptionPane.showMessageDialog(this,"Record added  successfully ");

                    ps.close();

                    con.close();

                } catch (Exception ex) {

                    ex.printStackTrace();

                }

            }

        });

    }

    void addC(){

        setLayout(new GridLayout(5,5));

        add=new JButton("  Add  ");

        add=new JButton("  Exit  ");

        eid=new JTextField(20);

        name=new JTextField(20);

        dept=new JTextField(20);

        salary=new JTextField(20);

        add(new JLabel("  emp id  "));

        add(eid);

        add(new JLabel("  name  "));

        add(name);

        add(new JLabel("  department  "));

        add(dept);

        add(new JLabel("  salary   "));

        add(salary);

        add(add);

        add(exit);

        pack();

    }

    public static void main(String[] args) {

                new RecordAdd().setVisible(true);

    }

}

**JDBC-Swing (Modified) => Practical5 =>Q.4**  
import java.awt.\*;  
import javax.swing.\*;  
import java.awt.event.\*;  
import java.sql.\*;  
  
public class RecordAdd extends JFrame implements ActionListener{  
  
    private JTextField name, eid, dept, salary;  
    private JButton add, exit;  
  
    public RecordAdd() {  
        setLayout(new GridLayout(5, 2));  
        add = new JButton("Add");  
        exit = new JButton("Exit");  
        eid = new JTextField(20);  
        name = new JTextField(20);  
        dept = new JTextField(20);  
        salary = new JTextField(20);  
        add(new JLabel(" emp id "));  
        add(eid);  
        add(new JLabel(" name "));  
        add(name);  
        add(new JLabel(" department "));  
        add(dept);  
        add(new JLabel(" salary "));  
        add(salary);  
        add(add);  
        add(exit);    
        add.addActionListener(this);  
        exit.addActionListener(this);  
        pack();  
         
    }  
  
    public static void main(String[] args) {  
        new RecordAdd().setVisible(true);  
    }  
  
    @Override  
    public void actionPerformed(ActionEvent ae) {  
        if(ae.getActionCommand().equals("Add" )) {  
        int id = Integer.parseInt(eid.getText());  
                String n = name.getText();  
                String d = dept.getText();  
                int s = Integer.parseInt(salary.getText());  
                Connection con;  
                PreparedStatement ps;  
                try {  
                    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb1","root","cs");  
                    ps = con.prepareStatement("INSERT INTO emp(empno, uname, dept, salary) VALUES( ?,  ?,  ?,  ?)");  
                    ps.setInt(1, id);  
                    ps.setString(2, n);  
                    ps.setString(3, d);  
                    ps.setInt(4, s);  
                    int i = ps.executeUpdate();  
                    if (i > 0) {                        
                     
                        JOptionPane.showMessageDialog(null, "Record added successfully");  
                    }  
                    ps.close();  
                    con.close();  
                } catch (Exception ex) {  
                    ex.printStackTrace();  
                }  
        }  
        else if (ae.getActionCommand().equals("Exit"))  
            System.exit(0);  
            }  
        }