**1) Java program to create class with instance variables,constructors,methods and clone the object.**

**Solution:**

```
class Book
{
	String name;
	int pages;
	double price;
	Book(String name,int pages,double price)
	{
		this.name=name;
		this.pages=pages;
		this.price=price;
		System.out.println("Book object created");
	}
	Book(Book b)
	{
		this.name=b.name;
		this.pages=b.pages;
		this.price=b.price;
	}
	void display()
	{
	System.out.println("Book details are");
	System.out.println("Book name="+name);
	System.out.println("Number of pages in the book="+pages);
	System.out.println("Price of the book="+price);
	}
}
public class TestBook
{
	public static void main(String[] args)
	{
		Book b1=new Book("java",700,1000);
		b1.display();
		Book b2=new Book(b1);
		System.out.println("After cloning...");
		b2.display();
	}
}
```

**Output:**

```
D:\lab\lab\p1>javac TestBook.java

D:\lab\lab\p1>java TestBook
Book object created
Book details are
Book name=java
Number of pages in the book=700
Price of the book=1000.0
After cloning...
Book details are
Book name=java
Number of pages in the book=700
Price of the book=1000.0
```

**2) Java program to illustrate the concept of dynamic method dispatch.**

**Solution:**

```
class A
{
   void test()
   {
      System.out.println("Invoking Test from A");
   }
}
class B extends A
{
   void test()
   {
      System.out.println("Invoking Test from B");
   }
}
class C extends B
{
   void test()
   {
      System.out.println("Invoking Test from C");
   }
}
public class TestDMD
{
   public static void main(String[] args)
   {
    A a=new A();
    B b=new B();
    C c=new C();
    A temp;
    temp=a;
    temp.test();
    temp=b;
    temp.test();
    temp=c;
    temp.test();
   }

}
```

**Output:**

```
D:\lab\lab\p2>javac TestDMD.java

D:\lab\lab\p2>java TestDMD
Invoking Test from A
Invoking Test from B
Invoking Test from C
```

**3) Java program to calculate salaries of n employees.**

**Solution:**

```
import java.util.*;
class Employee
{
   String name,designation;
   int id;
   double basic,da,hra,totalsalary;
   Scanner sc;
   public Employee()
   {
      sc=new Scanner(System.in);
      System.out.println("Enter employee name:");
      name=sc.next();
      System.out.println("Enter employee designation:");
      designation=sc.next();
      System.out.println("Enter employee id:");
      id=sc.nextInt();
      System.out.println("Enter employee basic pay:");
      basic=sc.nextDouble();
   }
   void calculatePay()
   {
      da=0.6*basic;
      hra=0.3*basic;
      totalsalary=basic+da+hra;
   }
   void getSalaryDetails()
   {
      System.out.println("Employee name="+name);
      System.out.println("Employee designation="+designation);
      System.out.println("Employee id="+id);
      System.out.println("Employee totalsalary="+totalsalary);
   }
}
public class TestEmployee
{
   public static void main(String[] args) {
      Scanner s=new Scanner(System.in);
      System.out.println("Enter number of employees:");
      int n=s.nextInt();
      Employee e[]=new Employee[n];
      for (int i = 0; i < e.length; i++)
```

```
        {
           System.out.println("Enter data for employee "+(i+1));
        e[i]=new Employee();
        e[i].calculatePay();
        }
        System.out.println("==============");
        for (int i = 0; i < e.length; i++)
        {
           System.out.println("Employee "+(i+1)+" details are:");
           e[i].getSalaryDetails();
        }
     }

}
```

**Output:**

```
D:\lab\lab\p3>javac TestEmployee.java

D:\lab\lab\p3>java TestEmployee
Enter number of employees:
2
Enter data for employee 1
Enter employee name:
john
Enter employee designation:
clerk
Enter employee id:
1234
Enter employee basic pay:
5000
Enter data for employee 2
Enter employee name:
smith
Enter employee designation:
manager
Enter employee id:
5678
Enter employee basic pay:
8000
==============
Employee 1 details are:
Employee name=john
Employee designation=clerk
Employee id=1234
Employee totalsalary=9500.0
Employee 2 details are:
Employee name=smith
Employee designation=manager
Employee id=5678
Employee totalsalary=15200.0
```

**4) Java program to perform various arithmetic operations using packages.**

**Solution:**

**File1:Addition.java**

```
package add;
public class Addition
{
  private int a,b;
  public Addition(int a,int b)
  {
        this.a=a;
        this.b=b;
  }
  public int addResult()
  {
        return a+b;
  }
}
```

**File2: Subtraction.java**

```
package subtract;
public class Subtraction
{
  private int a,b;
  public Subtraction(int a,int b)
  {
        this.a=a;
        this.b=b;
  }
  public int subResult()
  {
        return a-b;
  }
}
```

**File3:Product.java**

```
package add.subtract.multiply;
public class Product
{
  private int a,b;
  public Product(int a,int b)
  {
        this.a=a;
        this.b=b;
  }
```

```
  public int productResult()
  {
        return a*b;
  }
}
```

**File4:Division.java**

```
package div;
public class Division
{
  protected int a,b;
  public Division(int a,int b)
  {
        this.a=a;
        this.b=b;
  }
  public int divResult()
  {
        return a/b;
  }
}
```

**File5:ModuloDivision.java**

```
package div2;
import div.Division;
public class ModuloDivision extends Division
{
  public ModuloDivision(int a,int b)
  {
        super(a,b);
  }
  public int moduloResult()
  {
        return a%b;
  }
}
```

**File6:TestArithmetic.java**

```
import add.Addition;
import subtract.Subtraction;
import add.subtract.multiply.Product;
import div.Division;
import div2.ModuloDivision;
```

```
import java.util.Scanner;
public class TestArithmetic
{
  public static void main(String[] args)
  {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter two integers:");
        int k=sc.nextInt(),j=sc.nextInt();
        int result;
        Addition a=new Addition(k,j);
        result=a.addResult();
        System.out.println("Addition result="+result);
        Subtraction s=new Subtraction(k,j);
        result=s.subResult();
        System.out.println("Subtraction result="+result);
        Product p=new Product(k,j);
        result=p.productResult();
        System.out.println("Multiplication result="+result);
        Division d=new Division(k,j);
        result=d.divResult();
        System.out.println("Division result="+result);
        ModuloDivision md=new ModuloDivision(k,j);
        result=md.moduloResult();
        System.out.println("Modulo Divison result="+result);
  }
}
```

**Note:Compile File 1 to File 4 using "javac Filename.java -d ."**
**Compile file 5 using "javac TestArithmetic.java"**
**Output:**

```
D:\lab\lab\p4>javac Addition.java -d .

D:\lab\lab\p4>javac Subtraction.java -d .

D:\lab\lab\p4>javac Product.java -d .

D:\lab\lab\p4>javac Division.java -d .

D:\lab\lab\p4>javac ModuloDivision.java -d .

D:\lab\lab\p4>javac TestArithmetic.java

D:\lab\lab\p4>java TestArithmetic
Enter two integers:
8
5
Addition result=13
Subtraction result=3
Multiplication result=40
Division result=1
Modulo Divison result=3
```

**5) Java program design an ecommerce website using inheritance,abstract classes and dynamic polymorphism**

**Solution:**

```
import java.util.*;
interface OnlineShopping
{
   String items[]={"Laptops","Books","Mobiles","TV"};
   void viewProducts();
   void buyProducts();
}
interface BillPayments
{
   void payBill();
   String billType();
}
class Passenger
{
   String name,gender;
   int age;
   Passenger(String name, String gender, int age) {
      this.name = name;
      this.gender = gender;
      this.age = age;
   }
}
abstract class TrainTickets extends Passenger
{
   int ticketid;
   TrainTickets(String name,String gender,int age,int ticketid)
   {
      super(name, gender, age);
      this.ticketid=ticketid;
   }
   abstract void showTicketDetails();
}
class ECommerce extends TrainTickets implements OnlineShopping,BillPayments
{
   public ECommerce(String name, String gender, int age, int ticketid) {
      super(name, gender, age, ticketid);
   }
   void showTicketDetails() {
      System.out.println("Passenger name="+name);
      System.out.println("Gender="+gender);
```

```
        System.out.println("Age="+age);
        System.out.println("Ticketid="+ticketid);
    }
    public void viewProducts() {
        System.out.println("Available products are:");
        for(String k:items)
            System.out.println(k);
    }
    public void buyProducts() {
        System.out.println("Method to buy products");
    }
    public void payBill() {
        System.out.println("Method to pay bill");
    }
    public String billType() {
    return "Bill Type:MobileRecharge";
    }
}
public class TestEcommerce {
    public static void main(String[] args) {
        int choice;
        ECommerce ec=new ECommerce("john", "male", 23,12345);
        Scanner sc=new Scanner(System.in);
        do{
        System.out.println("1:TrainTicketing\n2:Products\n3:Bill payments\n4:Exit\nEnter choice:");
        choice=sc.nextInt();
            switch (choice)
            {
                case 1:
                    TrainTickets tt=ec;tt.showTicketDetails();
                    break;
                case 2:
                    OnlineShopping os=ec;os.viewProducts();os.buyProducts();
                    break;
                case 3:
                    BillPayments bp=ec;
                    System.out.println(bp.billType());
                    bp.payBill();
                    break;
                case 4:System.exit(0);
            }
        }while(choice!=5);
    }}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p5>javac TestEcommerce.java

D:\lab\lab\OoopUsingJava\src\p5>java TestEcommerce
1:TrainTicketing
2:Products
3:Bill payments
4:Exit
Enter choice:
1
Passenger name=john
Gender=male
Age=23
Ticketid=12345
1:TrainTicketing
2:Products
3:Bill payments
4:Exit
Enter choice:
2
Available products are:
Laptops
Books
Mobiles
TV
Method to buy products
1:TrainTicketing
2:Products
3:Bill payments
4:Exit
Enter choice:
4
```

**6) Java program to perform various String handling operations and stringtokenization.**

**Solution:**

```
import java.util.StringTokenizer;
public class StringDemo {
   public static void main(String[] args) {
      String s1="java";
      String s2="JAVA";
      System.out.println("s1="+s1);System.out.println("s2="+s2);
      System.out.println("s1==s2?"+s1.equals(s2));
      System.out.println("s1==s2?"+s1.equalsIgnoreCase(s2));
      System.out.println(s1.charAt(2)+","+s2.charAt(1));
      String s3=s1.concat("program");
      System.out.println("s3="+s3);
      System.out.println("CompareTo:"+s1.compareTo(s2));
      System.out.println("CompareTo:"+s1.compareToIgnoreCase(s2));
      System.out.println("Uppercase of s1="+s1.toUpperCase());
      System.out.println("Lowercase of s2="+s2.toLowerCase());
      System.out.println("s1 startswith="+s1.startsWith("j")+",s1 endswith="+s1.endsWith("a"));
      System.out.println("substring of s3="+s3.substring(2, s3.length()));
      char ch[]=new char[s3.length()];
      s3.getChars(0, 5, ch, 0);
      System.out.println(ch);
      String s4="This is a demo for string tokenization";
      System.out.println("s4="+s4);
      StringTokenizer token=new StringTokenizer(s4);
      System.out.println("Number of tokens/words="+token.countTokens());
      System.out.println("Tokens of '"+s4+"' are:");
      while (token.hasMoreTokens())
      {
         System.out.println(token.nextToken());
      }
   }
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p6>javac StringDemo.java

D:\lab\lab\OoopUsingJava\src\p6>java StringDemo
s1=java
s2=JAVA
s1==s2?false
s1==s2?true
v,A
s3=javaprogram
CompareTo:32
CompareTo:0
Uppercase of s1=JAVA
Lowercase of s2=java
s1 startswith=true,s1 endswith=true
substring of s3=vaprogram
javap
s4=This is a demo for string tokenization
Number of tokens/words=7
Tokens of 'This is a demo for string tokenization' are:
This
is
a
demo
for
string
tokenization
```

**7) Java program to handle multiple exceptions.**

**Solution:**

```
import java.util.*;
public class ExceptionDemo {
   public static void main(String[] args) {
      int a,b,c;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter two integers:");
      try{
      a=sc.nextInt();
      b=sc.nextInt();
         System.out.println("Division Result="+(a/b));
         System.out.println("Number of commandline arguments="+args.length);
         System.out.println("First argument="+args[0]);
      }
      catch(ArithmeticException ae)
      {
         System.out.println("Caught Division by zero error:"+ae);
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
         System.out.println("Exception caught="+e);

      }
      catch (Exception e)
      {
         System.out.println("Entered numbers are not integers");
      }
   }
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p7>javac ExceptionDemo.java

D:\lab\lab\OoopUsingJava\src\p7>java ExceptionDemo
Enter two integers:
5
0
Caught Division by zero error:java.lang.ArithmeticException: / by zero

D:\lab\lab\OoopUsingJava\src\p7>java ExceptionDemo
Enter two integers:
5
abc
Entered numbers are not integers

D:\lab\lab\OoopUsingJava\src\p7>java ExceptionDemo
Enter two integers:
8
4
Division Result=2
Number of commandline arguments=0
Exception caught=java.lang.ArrayIndexOutOfBoundsException: 0
```

**8) Java program to create and handle user defined exception.**

**Solution:**

```
class MyException extends Exception {
private int detail;
MyException(int a) {
detail = a;
}
public String toString() {
return "MyException[" + detail + "]";
}
}
public class UserDefinedException {
  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);
}
}
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p8>javac UserDefinedException.java

D:\lab\lab\OoopUsingJava\src\p8>java UserDefinedException
Called compute(1)
Normal exit
Called compute(20)
Caught MyException[20]
```

**9) Java program to demonstrate client-server environment using multithreading**

**Solution:**

```
import java.io.*;
class Data
{
   String msg;
   boolean status;
}
class Client extends Thread
{
   Data t;
   public Client(Data t) {
      this.t = t;
      t.status=false;
      start();
   }
   public void run()
   {
      stp:
      while (true)
      {
         while (!t.status)
         {
            try {
             BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
               System.out.println("CLIENT:Enter msg");
               String m=br.readLine();
               t.status=true;
               t.msg=m;
               if (t.msg.equals("stop"))
               {
               break stp;
               }
            } catch (Exception e) {}
         }
      }
   }
}
class Server extends Thread
{
   Data t;
   public Server(Data t) {
      this.t = t;
```

```
            start();
        }
        public void run()
        {
            stp:
            while (true)
            {
                while (t.status)
                {
                        System.out.println("SERVER:Msg received from client="+t.msg);
                        if (t.msg.equals("stop"))
                        break stp;
                        t.status=false;
                }
            }
        }

    public class ClientServer {
        public static void main(String[] args) {
            Data d=new Data();
            Client c=new Client(d);
            Server s=new Server(d);
        }
    }
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p9>java ClientServer
CLIENT:Enter msg
hai server
SERVER:Msg received from client=hai server
CLIENT:Enter msg
this is client thread
SERVER:Msg received from client=this is client thread
CLIENT:Enter msg
stop
SERVER:Msg received from client=stop
```

**10) Java program to perform mutual exclusion using thread synchronization**

**Solution:**

```
class CallMe
{
   synchronized void call(String msg)
   {
      System.out.print("["+msg);
      try {
         Thread.sleep(1000);
      } catch (Exception e) {}
      System.out.println("]");
   }
}
class Caller implements Runnable
{
   CallMe target;
   String msg;
   Thread t;
   public Caller(CallMe target,String msg) {
      this.msg = msg;
      this.target = target;
      this.t = new Thread(this);
      t.start();
   }
   public void run()
   {
      target.call(msg);
   }
}
public class SyncDemo {
   public static void main(String[] args) {
      CallMe cm=new CallMe();
      Caller obj1=new Caller(cm,"Hello");
      Caller obj2=new Caller(cm,"Synchronized");
      Caller obj3=new Caller(cm,"World");
      //wait for threads to finish
      try {
         obj1.t.join();obj2.t.join();obj3.t.join();
      } catch (Exception e) {}
   }
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p10>javac SyncDemo.java

D:\lab\lab\OoopUsingJava\src\p10>java SyncDemo
[Hello]
[World]
[Synchronized]
```

**11) Java program to demonstrate linkedlist,hashset classes and iterator interface**

**Solution:**

```
import java.util.*;
public class ListAndSetDemo {
   public static void main(String[] args) {
      LinkedList list=new LinkedList();
      System.out.println("List="+list+",size="+list.size());
      HashSet set=new HashSet();
      System.out.println("Set="+set+",size="+set.size());
      System.out.println("Adding elements to list...");

list.add("Ece");list.add("IT");list.add("EEE");list.addLast("Mechanical");list.add("Civil");list.addFirst("
Cse");
      System.out.println("List="+list+",size="+list.size());
      System.out.println("Adding elements to set...");
      set.add("Cse");
      set.add("Ece");set.add("IT");set.add("EEE");set.add("Mechanical");set.add("Civil");
      System.out.println("Set="+set+",size="+set.size());
      System.out.println("Iterating elements in the list:");
      Iterator it=list.iterator();
      while (it.hasNext())
         System.out.print(it.next()+"->");
      System.out.println("\nIterating elements in the set:");
      it=set.iterator();
      while (it.hasNext())
         System.out.println(it.next());
   }
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p11>javac ListAndSetDemo.java
Note: ListAndSetDemo.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\lab\lab\OoopUsingJava\src\p11>java ListAndSetDemo
List=[],size=0
Set=[],size=0
Adding elements to list...
List=[Cse, Ece, IT, EEE, Mechanical, Civil],size=6
Adding elements to set...
Set=[Civil, Cse, EEE, Ece, Mechanical, IT],size=6
Iterating elements in the list:
Cse->Ece->IT->EEE->Mechanical->Civil->
Iterating elements in the set:
Civil
Cse
EEE
Ece
Mechanical
IT
```

**12) Java program to demonstrate vector,enumeration and comparator interfaces**

**Solution:**

```
import java.util.*;
class Student
{
   String name;
   long rollno;
   public Student(String name, long rollno) {
      this.name = name;
      this.rollno = rollno;
   }
   public String toString() {
      return "Student{" + "name=" + name + ", rollno=" + rollno + "}";
   }
}
class NameSort implements Comparator<Student>
{
   public int compare(Student obj1, Student obj2) {
      return obj1.name.compareTo(obj2.name);
   }
}
class RollNoSort implements Comparator<Student>
{
   public int compare(Student obj1, Student obj2) {
    if(obj1.rollno>obj2.rollno)return 1;
    else if(obj1.rollno<obj2.rollno)return -1;
    return 0;
   }
}
public class VectorAndComparatorDemo {
   public static void main(String[] args) {
      Student s1=new Student("james", 2451);
      Student s2=new Student("herbert", 2452);
      Student s3=new Student("david", 1608);
      Vector v=new Vector();
      v.add(s1);v.add(s2);v.add(s3);
      System.out.println("Student Vector="+v);
      System.out.println("Sorting students through names");
      Collections.sort(v,new NameSort());
      System.out.println("After namesort,vector="+v);
      System.out.println("Sorting students through rollno");
      Collections.sort(v,new RollNoSort());
      System.out.println("After rollnosort,vector="+v);
```

```
            System.out.println("Enumerating vector elements...");
            Enumeration e=v.elements();
            while (e.hasMoreElements())
               System.out.println(e.nextElement());
         }
      }
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p12>javac VectorAndComparatorDemo.java
Note: VectorAndComparatorDemo.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\lab\lab\OoopUsingJava\src\p12>java VectorAndComparatorDemo
Student Vector=[Student{name=james, rollno=2451}, Student{name=herbert, rollno=2
452}, Student{name=david, rollno=1608}]
Sorting students through names
After namesort,vector=[Student{name=david, rollno=1608}, Student{name=herbert, r
ollno=2452}, Student{name=james, rollno=2451}]
Sorting students through rollno
After rollnosort,vector=[Student{name=david, rollno=1608}, Student{name=james, r
ollno=2451}, Student{name=herbert, rollno=2452}]
Enumerating vector elements...
Student{name=david, rollno=1608}
Student{name=james, rollno=2451}
Student{name=herbert, rollno=2452}
```

**13) Java program to accept data and display output in key,value pairs.**

**Solution:**

```
import java.util.*;
public class MapDemo
{
 static HashMap accept(String key,String value,HashMap hm)
 {
    HashMap m=hm;
    m.put(key, value);
    return m;
 }
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      HashMap map=new HashMap();
      System.out.println("Map="+map);
      System.out.println("Enter key:");
      String key=sc.next();
      System.out.println("Enter value:");
      String value=sc.next();
      map=accept(key, value, map);
      System.out.println("Map="+map);
      System.out.println("Enter one more key:");
       key=sc.next();
      System.out.println("Enter one more value:");
      value=sc.next();
      map=accept(key, value, map);
      System.out.println("Map="+map);
   }
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p13>javac MapDemo.java
Note: MapDemo.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\lab\lab\OoopUsingJava\src\p13>java MapDemo
Map={}
Enter key:
namekey
Enter value:
david
Map={namekey=david}
Enter one more key:
dateofbirthKey
Enter one more value:
20January1976
Map={namekey=david, dateofbirthKey=20January1976}
```

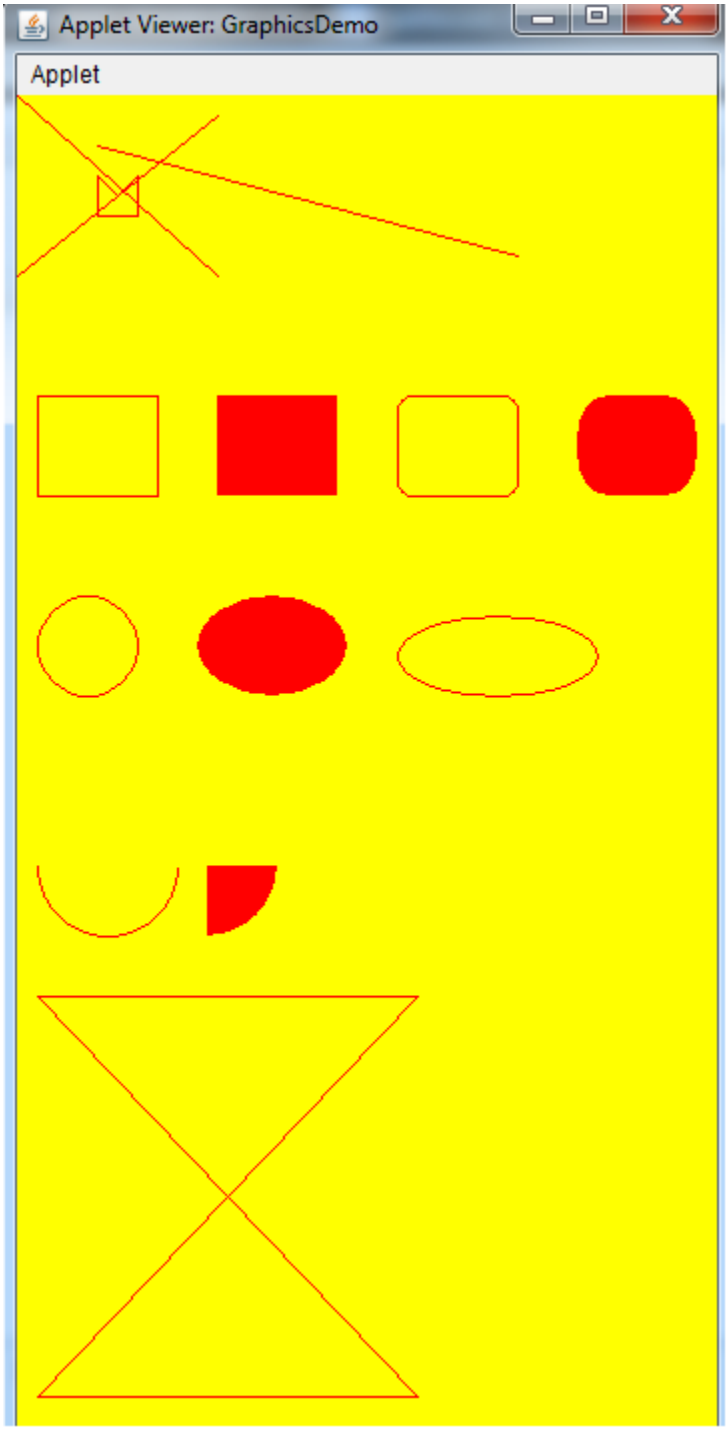
**14) Java program to demonstrate on Graphics class and insert various canvas objects on applet window.**

**Solution:**

```
import java.awt.*;
import java.applet.*;
import java.awt.color.ColorSpace;
/*
<applet code="GraphicsDemo" width=350 height=700>
</applet>
*/
public class GraphicsDemo extends Applet {
   public void init()
   {
      Color c1=Color.YELLOW;
      setBackground(c1);
      Color c2=Color.RED;
      setForeground(c2);
   }
public void paint(Graphics g) {
// Draw lines.
g.drawLine(0, 0, 100, 90);
g.drawLine(0, 90, 100, 10);
g.drawLine(40, 25, 250, 80);
// Draw rectangles.
g.drawRect(10, 150, 60, 50);
g.fillRect(100, 150, 60, 50);
g.drawRoundRect(190, 150, 60, 50, 15, 15);
g.fillRoundRect(280, 150, 60, 50, 30, 40);
// Draw Ellipses and Circles
g.drawOval(10, 250, 50, 50);
g.fillOval(90, 250, 75, 50);
g.drawOval(190, 260, 100, 40);
// Draw Arcs
g.drawArc(10, 350, 70, 70, 0, -180);
g.fillArc(60, 350, 70, 70, 0, -90);
// Draw a polygon
int xpoints[] = {10, 200, 10, 200, 10};
int ypoints[] = {450, 450, 650, 650, 450};
int num = 5;
g.drawPolygon(xpoints, ypoints, num);
int xmpoints[]={40,40,50,60,60};
int ympoints[]={60,40,50,40,60};
//Another polygon
g.drawPolygon(xmpoints, ympoints, num);
```

```
    }
    }
```

**Output:**




**Compile using:**
**javac GraphicsDemo.java**
**Run using:**
**appletviewer GraphicsDemo.java**

**15) Java awt program to demonstrate delegation event model.**

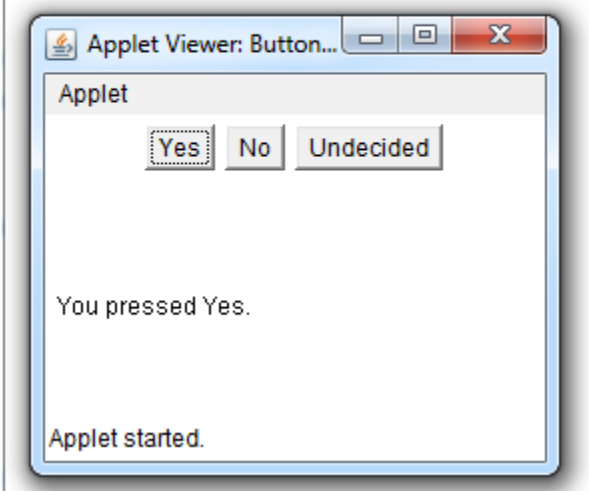
**Solution:**

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="ButtonDemo" width=250 height=150>
</applet>
*/
public class ButtonDemo extends Applet implements ActionListener {
String msg = "";
Button yes, no, maybe;
public void init() {
yes = new Button("Yes");
no = new Button("No");
maybe = new Button("Undecided");
add(yes);
add(no);
add(maybe);
yes.addActionListener(this);
no.addActionListener(this);
maybe.addActionListener(this);
}
public void actionPerformed(ActionEvent ae) {
String str = ae.getActionCommand();
if(str.equals("Yes")) {
msg = "You pressed Yes.";
}
else if(str.equals("No")) {
msg = "You pressed No.";
}
else {
msg = "You pressed Undecided.";
}
repaint();
}
public void paint(Graphics g) {
g.drawString(msg, 6, 100);
}
}
```

**Output:**

```
D:\lab\lab\p15>javac ButtonDemo.java

D:\lab\lab\p15>appletviewer ButtonDemo.java
```



**16) Java program to demonstrate on MouseListener.**

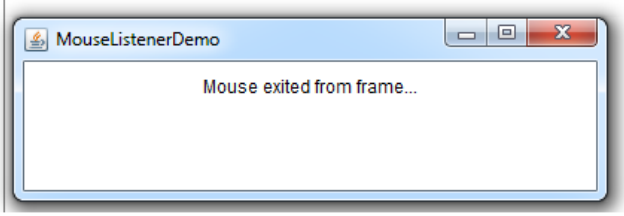
**Solution:**

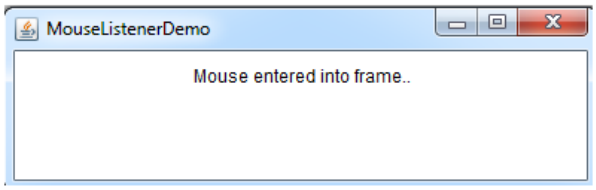
```
import java.awt.*;
import java.awt.event.*;
public class MouseListenerDemo extends Frame implements MouseListener
{
   Label l;
   public MouseListenerDemo(String title)
   {
      super(title);
      l=new Label("This Label is from frame");
      add(l);
      setLayout(new FlowLayout());
      addMouseListener(this);
      addWindowListener(new WindowAdapter()
      {
         public void windowClosing(WindowEvent we)
         {
            System.exit(0);
         }
      });
      setSize(400,500);
      setVisible(true);
   }
   public void mouseClicked(MouseEvent e)
   {
         l.setText("Mouse clicked...");
   }
   public void mousePressed(MouseEvent e)
   {
      l.setText("Mouse Pressed...");
   }
   public void mouseReleased(MouseEvent e)
   {
      l.setText("Mouse Released at..."+e.getX()+","+e.getY());
   }
   public void mouseEntered(MouseEvent e)
   {
      l.setText("Mouse entered into frame...");
   }
   public void mouseExited(MouseEvent e)
   {
      l.setText("Mouse exited from frame...");
```

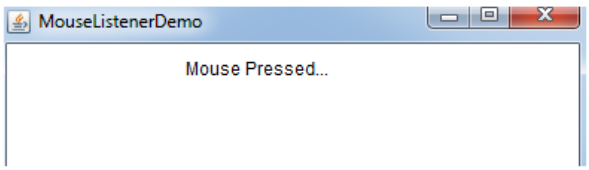
```
    }
    public static void main(String[] args) {
       MouseListenerDemo mld=new MouseListenerDemo("MouseListenerDemo");
    }
}
```

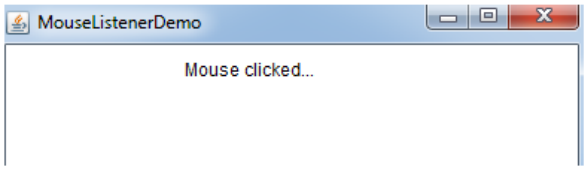
**Output:**

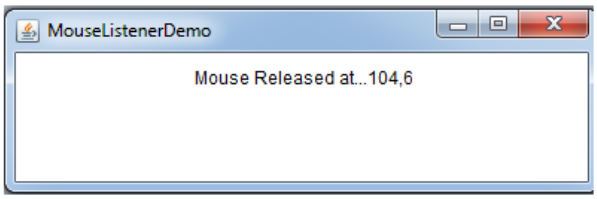
```
D:\lab\lab\p16>javac MouseListenerDemo.java
D:\lab\lab\p16>java MouseListenerDemo
```



**17) Java program to create a registration form with different controls,menus and event Handling.**

**Solution:**

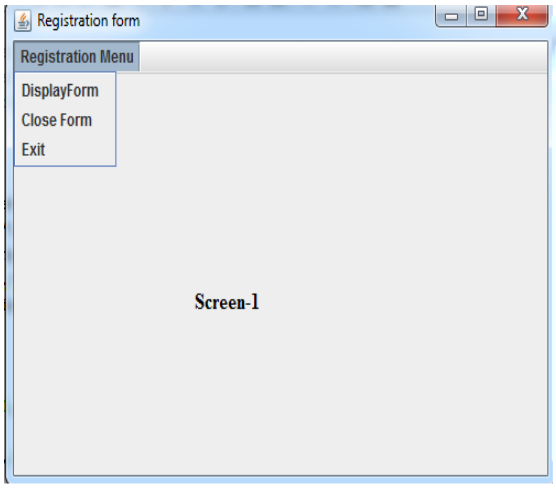
```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
import java.util.*;
public class RegistrationForm  extends JFrame implements ActionListener{

   JPanel p1;
   JMenu m;
   JMenuItem mi1,mi2,mi3;
   JMenuBar mb;
   JLabel l1,l2,l3,l4,l5,l6,l7,l8;
   JButton b1;
   JTextField tf1,tf2;
   JComboBox cb1,cb2,cb3,cb4;ButtonGroup bg;
   JRadioButton rb1,rb2;
   JTextArea ta;
   Object course[]={"B.E","M.Tech"},branch[]={"CSE","IT","ECE","EEE"};
   Object year[]={"1","2","3","4"},sem[]={"1","2"};
   public RegistrationForm(String title) {
      super(title);
      p1=new JPanel(new FlowLayout(10, 1, 2));
      mb=new JMenuBar();
      mi1=new JMenuItem("DisplayForm");
      mi2=new JMenuItem("Close Form");
      mi3=new JMenuItem("Exit");
      m=new JMenu("Registration Menu");
      m.add(mi1);m.add(mi2);m.add(mi3);mb.add(m);setJMenuBar(mb);
      mi1.addActionListener(this);mi2.addActionListener(this);mi3.addActionListener(this);
      l1=new JLabel("Enter name:");l2=new JLabel("Select course:");
      l3=new JLabel("Select branch:");
      l4=new JLabel("Year:");l5=new JLabel("Sem:");l6=new JLabel("Enter Roll number:");
      l7=new JLabel("Gender:");l8=new JLabel("Enter address:");
      tf1=new JTextField(10);tf2=new JTextField(15);ta=new JTextArea(5,10);
     cb1=new JComboBox(course);cb2=new JComboBox(branch);
     cb3=new  JComboBox(year);cb4=new JComboBox(sem);
    rb1=new JRadioButton("Male");rb2=new JRadioButton("Female");
    bg=new ButtonGroup();bg.add(rb1);bg.add(rb2);
    b1=new JButton("Register");
    b1.addActionListener(this);
     p1.add(l1);p1.add(tf1);p1.add(l2);p1.add(cb1);p1.add(l3);
```

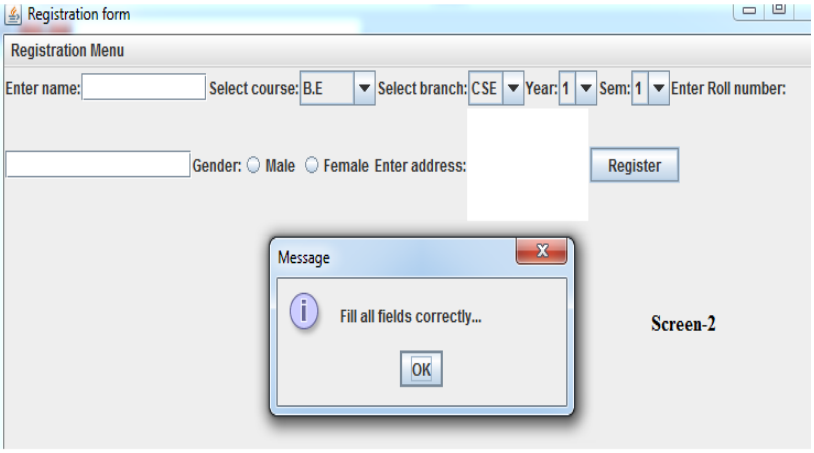
```
        p1.add(cb2);p1.add(l4);p1.add(cb3);
        p1.add(l5);p1.add(cb4);p1.add(l6);p1.add(tf2);p1.add(l7);p1.add(rb1);p1.add(rb2);
        p1.add(l8);p1.add(ta);p1.add(b1);
        add(p1);p1.setVisible(false);
        setSize(500,500);setVisible(true);setDefaultCloseOperation(3);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource()==mi1) p1.setVisible(true);
        else if(ae.getSource()==mi2)p1.setVisible(false);
        else if(ae.getSource()==mi3)System.exit(0);
        else
        {
            if(tf1.getText().equals("") || tf2.getText().equals("") || ta.getText().equals(""))
            JOptionPane.showMessageDialog(rootPane, "Fill all fields correctly...");
            else
            JOptionPane.showMessageDialog(rootPane, "Registration Form filled successfully...");
        }
    }
    public static void main(String[] args) {
        new RegistrationForm("Registration form");
    }
}
```

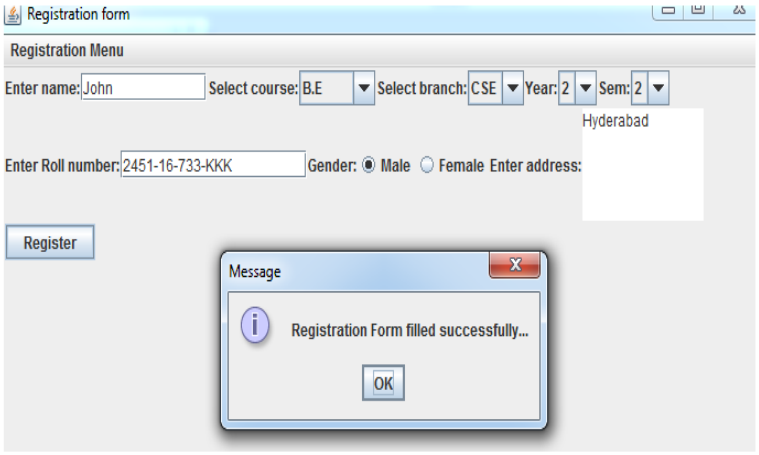
## Output:




Screen-1




Screen-2




Screen-3

**18) Java program to create a login form with necessary controls,layout managers and dialog boxes using swings.**

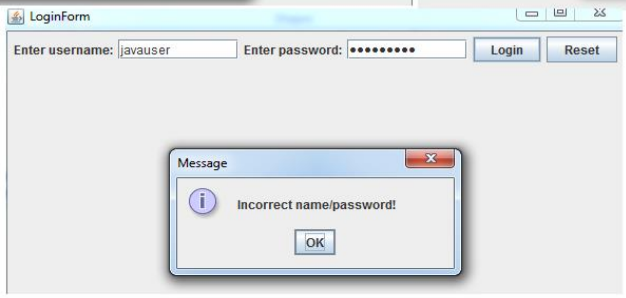
**Solution:**

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class LoginForm extends JFrame implements ActionListener{
JLabel l1,l2;
JTextField tf;
JPasswordField pf;
JButton b1,b2;
   public LoginForm(String title) {
      super(title);
      l1=new JLabel("Enter username:");
      l2=new JLabel("Enter password:");
      tf=new JTextField(10);
      pf=new JPasswordField(10);
      b1=new JButton("Login");b2=new JButton("Reset");
      b1.addActionListener(this);b2.addActionListener(this);
      setLayout(new FlowLayout());
      add(l1);add(tf);add(l2);add(pf);add(b1);add(b2);
      setSize(600,400);
      setVisible(true);
   }
   public void actionPerformed(ActionEvent e) {
      if (e.getSource()==b1) {
         String name=tf.getText(),pwd=pf.getText();
         if (name.equals("javauser")&&pwd.equals("java1234")) {
            JOptionPane.showMessageDialog(rootPane, "Login success..!");
         } else if(name.equals("")||pwd.equals("")) {
            JOptionPane.showMessageDialog(rootPane, "Fill all fields..!");
         }else JOptionPane.showMessageDialog(rootPane, "Incorrect name/password!");
      } else {
         tf.setText("");pf.setText("");
      }
   }
   public static void main(String[] args) {
      new LoginForm("LoginForm");
   }
}
```

## Output:


Screen-1


Screen-2




Screen-3

**19) Java program to design a simple calculator using swings and apply event handlers.**

**Solution:**

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class Calc extends JFrame implements ActionListener
{
   JTextField tf;
   JButton b[]=new JButton[10];
   JButton add,sub,eq;
   JPanel p1,p2;
int num1,num2,result;
char ch,optr;
   public Calc(String title)
   {
      super(title);
    Container c=getContentPane();
    c.setLayout(new FlowLayout());
    p1=new JPanel();p2=new JPanel(new GridLayout(4,4));
      tf=new JTextField(10);
      for (int i = 0; i < 10; i++) {
          b[i]=new JButton(i+"");
      }
      add=new JButton("+");
      sub=new JButton("-");
      eq=new JButton("=");
     p1.add(tf);
     for (int i = 0; i < 10; i++)
     {
        p2.add(b[i]);
         b[i].addActionListener(this);
     }
      p2.add(add);
      add.addActionListener(this);
      p2.add(sub);
      sub.addActionListener(this);
      p2.add(eq);
      eq.addActionListener(this);
    add(p1);add(p2);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setSize(600,600);
      setVisible(true);
   }
```

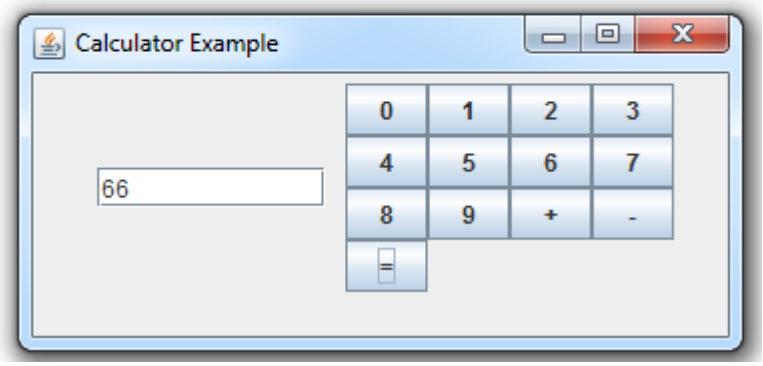
```
    public void actionPerformed(ActionEvent e)
    {
       String str=e.getActionCommand();
       ch=str.charAt(0);
       if (Character.isDigit(ch))
       {
          tf.setText(tf.getText()+str);

       } else
       {
          if(str.equals("+"))
          {
             optr='+';
             num1=Integer.parseInt(tf.getText());
             tf.setText("");
          }else if(str.equals("-"))
          {
             optr='-';
             num1=Integer.parseInt(tf.getText());
             tf.setText("");
          }
          else if(str.equals("="))
          {
             num2=Integer.parseInt(tf.getText());
             switch (optr)
             {
                case '+':result=num1+num2;
                   break;
                case '-':result=num1-num2;
                   break;
             }
             tf.setText(result+"");
          }
       }
    }
    public static void main(String[] args) {
       new Calc("Calculator Example");
    }
}
```

**Output:**

```
D:\lab\lab\p19>javac Calc.java

D:\lab\lab\p19>java Calc
```



**20) Java program to copy data from one file to another file.**

**Solution:**

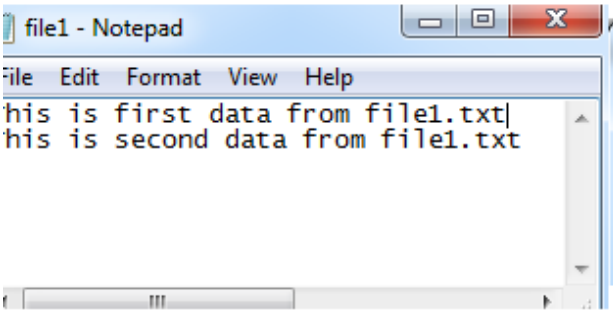
```
import java.io.*;
import java.util.*;
public class FileCopy
{
   public static void main(String[] args) {
      String srcfile,destfile;
      int k;
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter source file with extension");
      srcfile=sc.next();
      System.out.println("Enter destination file with extension to copy");
      destfile=sc.next();
      try{
      FileReader fr=new FileReader(srcfile);
      FileWriter fw=new FileWriter(destfile);
         while ((k=fr.read())!=-1)
         {
            fw.write(k);
         }
         System.out.println("File content copied...");
         fr.close();fw.close();
      }catch(Exception e)
      {
         System.out.println("Source File not found...");
      }
   }

}
```

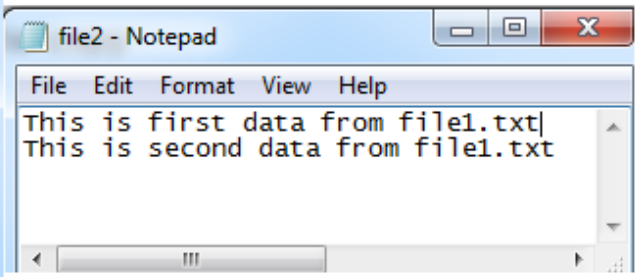
**Output:**

```
D:\lab\lab\OoopUsingJava\src\p20>javac FileCopy.java

D:\lab\lab\OoopUsingJava\src\p20>java FileCopy
Enter source file with extension
file1.txt
Enter destination file with extension to copy
file2.txt
File content copied...
```




**Input file**




**Outputfile**

**21) Java program to merge contents of two files and display output on console.**

**Solution:**

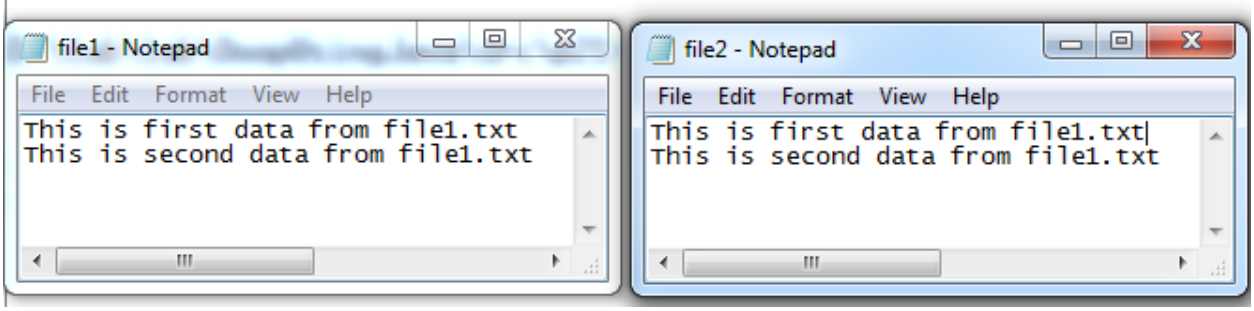
**Note: First create two files file1.txt and file2.txt and execute this program**

```
import java.io.*;
public class FileMerge
{
   public static void main(String[] args) {
      try {
         InputStream fis1=new FileInputStream("file1.txt");
         InputStream fis2=new FileInputStream("file2.txt");
         InputStream is=new SequenceInputStream(fis1, fis2);
         int k;
         System.out.println("Data after merging contents\n======");
         while ((k=is.read())!=-1)
            System.out.print((char)k);
         System.out.println("Merge completed");
         fis1.close();fis2.close();
         is.close();
      } catch (Exception e) {
         System.out.println("Input files not found..");
      }
   }
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p21>javac FileMerge.java

D:\lab\lab\OoopUsingJava\src\p21>java FileMerge
Data after merging contents
======
This is first data from file1.txt
This is second data from file1.txt
This is first data from file1.txt
This is second data from file1.txt
Merge completed
```



**22) Java program to illustrate serialization**

**Solution:**

```
import java.io.*;
class Student implements Serializable
{
   String name;
   long rollno;
   public Student(String name, long rollno) {
      this.name = name;
      this.rollno = rollno;
   }
   public String toString() {
      return "Student{" + "name=" + name + ", rollno=" + rollno + '}';
   }
}

public class SerializeDemo {
   public static void main(String[] args) {
      Student s1=new Student("student1", 2451);
      Student s2=new Student("student2",2452);
      Student s3=new Student("student3",1603);
      try{
      FileOutputStream fos=new FileOutputStream("serialize.txt");
      ObjectOutputStream oos=new ObjectOutputStream(fos);
      oos.writeObject(s1);oos.writeObject(s2);oos.writeObject(s3);
      System.out.println("File serialization completed...");
      oos.close();fos.close();
      System.out.println("Performing deserialization\n======");
      FileInputStream fis=new FileInputStream("serialize.txt");
      ObjectInputStream ois=new ObjectInputStream(fis);
      Student ds1=(Student)ois.readObject();
      Student ds2=(Student)ois.readObject();
      Student ds3=(Student)ois.readObject();
      System.out.println(ds1);System.out.println(ds2);System.out.println(ds2);
      ois.close();fis.close();
      }catch(Exception e)
      {
         System.out.println("Serialization not possible...");
      }
   }
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p22>javac SerializeDemo.java

D:\lab\lab\OoopUsingJava\src\p22>java SerializeDemo
File serialization completed...
Performing deserialization
======
Student{name=student1, rollno=2451}
Student{name=student2, rollno=2452}
Student{name=student2, rollno=2452}
```

**23) Java program to retrieve webpage using URL class.**

**Solution:**

```
import java.io.*;
import java.net.*;
import java.util.*;
public class URLDemo {
   public static void main(String[] args) throws Exception{
      String str;
      System.out.println("Enter url name:");
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      str=br.readLine();
      URL url=new URL(str);
      System.out.println("Protocol="+url.getProtocol());
      System.out.println("Host="+url.getHost());
      System.out.println("File="+url.getFile());
      System.out.println("Path="+url.getPath());
      System.out.println("Port="+url.getDefaultPort());
     br.close();
     URLConnection conn=url.openConnection();
     br=new BufferedReader(new InputStreamReader(conn.getInputStream()));
     String temp;
      while ((temp=br.readLine())!=null) {
         System.out.println(temp);
         //br.close();
      }
   }
}
```

**Output:**

```
D:\lab\lab\OoopUsingJava\src\p23>javac URLDemo.java

D:\lab\lab\OoopUsingJava\src\p23>java URLDemo
Enter url name:
https://www.google.com/
Protocol=https
Host=www.google.com
File=/
Path=/
Port=443
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-
IN"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><me
ta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itempr
op="image"><title>Google</title><script nonce="b6aMaMh9Aq5sNuHsLQYekQ==">(functi
on(){window.google={kEI:'yJrIWvwPyve8BPf3tsgE',kEXPI:'0,1353747,529,640,759,86,1
31,548,339,16,58,366,336,342,39,176,109,5,51,123,2341918,122,81,329294,1294,1238
3,2347,2508,32691,16115,769,7,792,7,661,4456,5471,6381,3101,234,2,2,1624,514,246
7,2196,367,550,332,332,326,1776,113,2201,19,2114,1058,224,2212,136,49,81,3718,13
,248,210,439,479,444,1046,750,1,7,6,20,353,24,287,64,134,176,3,299,432,618,402,4
3,324,37,29,2,2,2,269,358,51,612,388,789,965,147,300,133,712,5,154,323,3,95,309,
129,274,20,132,365,695,1,51,429,38,7,3,41,44,516,15,630,8,326,21,191,770,66,196,
722,137,441,706,365,199,409,3,47,218,57,2,184,167,84,324,2,107,7,249,596,325,42,
```

**24) Java program to load and display image and perform gray scale.**

**Solution:**

**Given input image name=”Tulips.jpg”**

**Output image name=”output.jpg”**

```
import java.io.File;
import java.io.IOException;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;

public class ImageGrayScale{
  public static void main(String args[])throws IOException{
    BufferedImage img = null;
    File f = null;
    //read image
    try{
      f = new File("/home/mvsr/Tulips.jpg");
      img = ImageIO.read(f);
    }catch(IOException e){
      System.out.println(e);
    }
    //get image width and height
    int width = img.getWidth();
    int height = img.getHeight();
    //convert to grayscale
    for(int y = 0; y < height; y++){
      for(int x = 0; x < width; x++){
        int p = img.getRGB(x,y);
        int a = (p>>24)&0xff;
        int r = (p>>16)&0xff;
        int g = (p>>8)&0xff;
        int b = p&0xff;
        //calculate average
        int avg = (r+g+b)/3;
        //replace RGB value with avg
        p = (a<<24) | (avg<<16) | (avg<<8) | avg;
        img.setRGB(x, y, p);
      }
    }
    //write image
    try{
      f = new File("/home/mvsr/output.jpg");
      ImageIO.write(img, "jpg", f);
    }catch(IOException e){
```

```
        System.out.println(e);
      }
    }
}
```

**Output:**

Input colored image,Tulips.jpg

Output grayscale image(black&white),Tulips.jpg

**25) Java program to demonstrate message communication using sockets.**

**Solution:**

**File1 : Socket1.java**

```
import java.util.*;
import java.io.*;
import java.net.*;
public class Socket1
{
	public static void main(String[] args)throws Exception
	{
		System.out.println("Socket1 is ready...");
		ServerSocket ss=new ServerSocket(1234);
		Socket s=ss.accept();
		Scanner sc=new Scanner(System.in);
		DataOutputStream dos=new DataOutputStream(s.getOutputStream());
		DataInputStream dis=new DataInputStream(s.getInputStream());
		String msg1="",msg2="";
		while(!msg1.equals("quit"))
		{
			msg1=dis.readUTF();
			System.out.println("Message from Socket2="+msg1);
			msg2=sc.nextLine();
			dos.writeUTF(msg2);
		}
		dis.close();
	}
}
```

**File2 : Socket2.java**

```
import java.util.*;
import java.io.*;
import java.net.*;
public class Socket2
{
	public static void main(String[] args)
	{
		System.out.println("Trying to connect to socket2...");
		try
		{
			Socket s=new Socket("localhost",1234);
			System.out.println("Successfully Connected to socket1...");
	System.out.println("Now start communication...Type 'quit' to stop communication");
			Scanner sc=new Scanner(System.in);
```

```
                DataOutputStream dos=new DataOutputStream(s.getOutputStream());
                DataInputStream dis=new DataInputStream(s.getInputStream());
                String msg1="",msg2="";
                System.out.println("");
                while(!msg1.equals("quit"))
                {
                        msg1=sc.nextLine();
                        dos.writeUTF(msg1);
                        msg2=dis.readUTF();
                        System.out.println("Message from Socket1="+msg2);
                }
                dos.close();
            }catch(Exception e)
            {
        System.out.println("Socket1 is not running...you must run socket 1 program first..!");
            }
        }
    }
```

**Output:**

**Note: Before compilation,create a folder "socket" and place two java files in this "socket" folder and compile using the following command:**

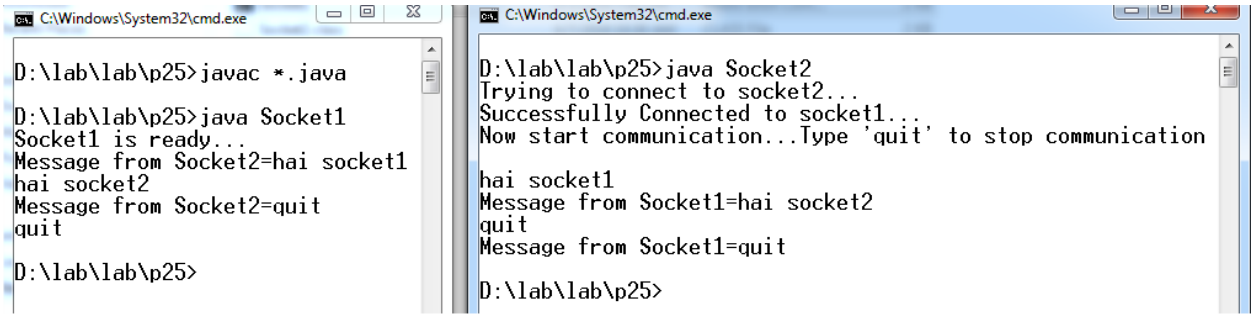
**javac *.java**

**First, open one LXTerminal and run Socket1 using**

**java Socket1**

**Second, open one more LXTerminal and run Socket2 using**

**java Socket2**




**Terminal 1** **Terminal2**