1. SAMPLE PROGRAM EMPLOYING COMPLEX NUMBER

```
import java.io.*;
class complex1
{
   int real,imag;
   DataInputStream obj=new DataInputStream(System.in);
   public void read() throws IOException
   {
      System.out.println("Enter the real part");
      real=Integer.parseInt(obj.readLine());
      System.out.println("Enter the imaginary part");
      imag=Integer.parseInt(obj.readLine());
   }
public void add(complex1 c1,complex1 c2)throws IOException
   {
   real=c1.real+c2.real;
   imag=c1.imag+c2.imag;
   }
public void display()throws IOException
   {
   System.out.println(real+"+"+imag+"i");
   }
}
public class complex
   {
   public static void main(String args[]) throws IOException
      {
      complex1 x=new complex1();
      complex1 y=new complex1();
      complex1 z=new complex1();
      System.out.println("Enter the first number");
      x.read();
      System.out.println("Enter the second number");
      y.read();
      z.add(x,y);
      System.out.println("first number");
      x.display();
      System.out.println("second number");
      y.display();
      System.out.println("sum:");

    z.display();
      }}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano complex.java
root@osboxes:/home/osboxes# javac complex.java
Note: complex.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
root@osboxes:/home/osboxes# java complex
Enter the first number
Enter the real part
32
Enter the imaginary part
23
Enter the second number
Enter the real part
13
Enter the imaginary part
29
first number
32+23i
second number
13+29i
sum:
45+52i
root@osboxes:/home/osboxes#
```

2. NESTED AND INNER CLASS

```
import java.io.*;
public class nested
{
 public int x=0;
 class inner
 {
  public int x=1;
  void methodIninner(int x)
  {
    System.out.println("x="+x);
    System.out.println("this.x="+this.x);
    System.out.println("nested.this.x="+nested.this.x);
  }
}
public static void main(String args[])
{
    nested obj=new nested();
    nested.inner obj1=obj.new inner();
    obj1.methodIninner(45);
 }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano nested.java
root@osboxes:/home/osboxes# javac nested.java
root@osboxes:/home/osboxes# java nested
x=45
this.x=1
nested.this.x=0
root@osboxes:/home/osboxes#
```

3. STRING MANIPULATION

```
import java.io.*;
import java.util.*;
class std1
{
  public static void main(String args[]) throws IOException
  {
    String s1,s2,s3,s4,s;
    int a,m,n;
    Scanner sc =new Scanner (System.in);
    do
    {
      System.out.println("\nMain menu\n\t 1:Length of the string\n\t 2:Substring\n\t
3:Compare\n\t 4:UpperCase\n\t 5:Concatenation\n\t 6:Exit");
      System.out.println("Enter your choice:");
      a=sc.nextInt();
      switch(a)
      {
        case 1:
          System.out.println("Enter the string:");
          s=sc.next();
          System.out.println("\tThe length of the string:" +s.length());
          break;
        case 2:
          System.out.println("Enter the string:");
          s1=sc.next();
          System.out.println("Enter the start index:");
          m=sc.nextInt();
          System.out.println("Enter the end index:");
          n=sc.nextInt();
          s4=s1.substring(m,n);
          System.out.println("Substring is:" +s4);
          break;
        case 3:
          System.out.println("Enter two strings:");
          s1=sc.next();
          s2=sc.next();
          if(s1.compareTo(s2)<0)
            System.out.println("String" +s1+" is less than" +s2);
          else if(s1.compareTo(s2)>0)
            System.out.println("String" +s1+" is greater than"+s2);

        else
```

```
                System.out.println("String are equal");
                break;

            case 4:
                System.out.println("Enter the string:");
                s1=sc.next();
                s2=s1.toUpperCase();
                System.out.println("String in uppercase:"+s2);
                break;
            case 5:
                System.out.println("Enter two strings:");
                s1=sc.next();
                s2=sc.next();
                s3=s1.concat(s2);
                System.out.println("Concatenated String is:" +s3);
                break;
            case 6:
                System.exit(0);
            default:
                System.out.println("Wrong choice");
        }
    } while(a!=6);
  }

}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano std1.java
root@osboxes:/home/osboxes# javac std1.java
root@osboxes:/home/osboxes# java std1

Main menu
         1:Length of the string
         2:Substring
         3:Compare
         4:UpperCase
         5:Concatenation
         6:Exit
Enter your choice:
1
Enter the string:
environment
        The length of the string:11

Main menu
         1:Length of the string
         2:Substring
         3:Compare
         4:UpperCase
         5:Concatenation
         6:Exit
Enter your choice:
2
```

```
Enter your choice:
2
Enter the string:
environment
Enter the start index:
2
Enter the end index:
8
Substring is:vironm

Main menu
         1:Length of the string
         2:Substring
         3:Compare
         4:UpperCase
         5:Concatenation
         6:Exit
Enter your choice:
3
Enter two strings:
hii
hii
String are equal
```

```
Main menu
        1:Length of the string
        2:Substring
        3:Compare
        4:UpperCase
        5:Concatenation
        6:Exit
Enter your choice:
4
Enter the string:
hello
String in uppercase:HELLO

Main menu
        1:Length of the string
        2:Substring
        3:Compare
        4:UpperCase
        5:Concatenation
        6:Exit
Enter your choice:
5
Enter two strings:
hello
world
Concatenated String is:helloworld
```

4. COMMAND LINE ARGUMENT(SUM OF NUMBERS)

```
import java.io.*;
class command
 {
 public static void main(String args[])
  {
     int n=0;
     for(int i=0;i<args.length;i++)
     {
       n=n+Integer.parseInt(args[i]);
     }
     System.out.println("The sum is:"+n);
     }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano command.java
root@osboxes:/home/osboxes# javac command.java
root@osboxes:/home/osboxes# java command 1 2 3 4
The sum is:10
root@osboxes:/home/osboxes#
```

5. A) SINGLE INHERITANCE

```
class A
{
 void display()
 {
  System.out.println("hello");
 }
}
public class single extends A
{

 public static void main(String args[])
 {
  single b=new single();
  b.display();
 }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano single.java
root@osboxes:/home/osboxes# javac single.java
root@osboxes:/home/osboxes# java single
hello
root@osboxes:/home/osboxes#
```

5.B) MULTILEVEL INHERITANCE

```
import java.io.*;
import java.util.*;
class student
{
   int rno,m1,m2,m3;
   float avg,total;
   String sname;
   Scanner sc=new Scanner(System.in);
   public void read()
   {
      System.out.println("enter the details");
      System.out.println("enter the id");
      rno=sc.nextInt();
      System.out.println("enter the name");
      sname=sc.next();
      System.out.println("enter the marks");
      m1=sc.nextInt();
      m2=sc.nextInt();
      m3=sc.nextInt();
   }
}
class mark extends student
{
    public void calc()
    {
       total=m1+m2+m3;
       avg=(total/150)*100;
    }
   public void display()
   {
      System.out.println("sno"+"\t"+"sname"+"\t"+"mark1"+"\t"+"mark2"+"\t"+"m
ark3"+"\t"+"Total"+"\t"+"Average");
      System.out.println(rno+"\t"+sname+"\t"+m1+"\t"+m2+
      "\t"+m3+"\t"+total+"\t "+avg);
   }
}
class multilevel extends mark
{
   public static void main(String args[])
   {

    multilevel m=new multilevel();
     m.read();
```

```
        m.calc();
        m.display();
    }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano nutlilevel.java
root@osboxes:/home/osboxes# javac multilevel.java
root@osboxes:/home/osboxes# java multilevel
enter the details
enter the id
100
enter the name
rihan
enter the marks
98
89
76
sno     sname   mark1   mark2   mark3   Total   Average
100     rihan   98      89      76      263.0    175.33333
root@osboxes:/home/osboxes#
```

5.C) MULTIPLE INHERITANCE

```
import java.io.*;
interface idemo
{
   void show();
}
class first
{
   void display()
{
   System.out.println("show first");
}
}
class second extends first implements idemo
{
   public void show()
   {
   System.out.println("show of second");
   }
}
class multiple
{
   public static void main(String args[])
   {
      second ref=new second();
      ref.show();
      ref.display();
   }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano mutliple.java
root@osboxes:/home/osboxes# javac multiple.java
root@osboxes:/home/osboxes# java multiple
show of second
show first
root@osboxes:/home/osboxes#
```

5.D) HYBRID INHERITANCE

```
import java.io.*;
class C
{
	public void display()
	{
		System.out.println("Super class");
	}
}
class A extends C
{
	public void display()
	{
		System.out.println("Derived class of C");
		super.display();
	}
}
class B extends C
{
	public void display()
	{
	System.out.println("Derived class of C");
	super.display();
	}
}
class hybrid extends A
{
	public void display()
	{
	System.out.println("Derived class of A");
	super.display();
	}
	public static void main(String args[])
	{
	System.out.println("combination of single and hierarchical inheritance:");
	hybrid d=new hybrid();
	d.display();
	}
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano hybrid.java
root@osboxes:/home/osboxes# javac hybrid.java
root@osboxes:/home/osboxes# java hybrid
combination of single and hierarchical inheritance:
Derived class of A
Derived class of C
Super class
root@osboxes:/home/osboxes#
```

5.E) METHOD OVERLOAD

```
import java.io.*;
public class overload
{
 public int sum(int x,int y)
 {
  return(x+y);
 }
 public int sum(int x,int y,int z)
 {
  return(x+y+z);
 }
 public double sum(double x,double y)
 {
  return(x+y);
 }
 public string sum(string x,string y)
 {
  return(x+y);
 }
 public static void main(String args[])
 {
   overload s = new overload();
   System.out.println(s.sum(10,20));
   System.out.println(s.sum(10,20,30));
   System.out.println(s.sum(10.54,20.50));
   System.out.println(s.sum("kannur","university"));
 }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano overloads.java
root@osboxes:/home/osboxes# javac overloads.java
root@osboxes:/home/osboxes# java overloads
3
6
43.0
kannuruniversity
root@osboxes:/home/osboxes#
```

5.F) METHOD OVERRIDE

```
import java.io.*;
class parent
{
   public void m1()
   {
      System.out.println("From parent m1()");
   }
}
public class override extends parent
{
   public void m1()
   {
      System.out.println("From child m1()");
      super.m1();
   }
   public static void main(String args[])
   {
      override ob1=new override();
      ob1.m1();
   }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano override.java
root@osboxes:/home/osboxes# javac override.java
root@osboxes:/home/osboxes# java override
From child m1()
From parent m1()
root@osboxes:/home/osboxes#
```

6.PACKAGES
A) DISPLAY DETAILS OF AN ITEM

```
//FILE NAME item.java

package p1;
import java.util.*;
public class item
{
  int quantity,price,id;
  String name;
  Scanner sc=new Scanner(System.in); public
  void read()
  {
    System.out.println("enter the item name"); name=sc.next();
    System.out.println("enter the item id"); id=sc.nextInt();
    System.out.println("enter the item quantity"); quantity=sc.nextInt();
    System.out.println("enter the item price"); price=sc.nextInt();
  }
  public void display()
  {
    System.out.println("Item name\tItem id\t\tItem quantity\tItem price\t");
    System.out.println(name+"\t\t"+id+"\t\t"+quantity+"\t\t"+price+"\t");
  }
}


//*FILE NAME item1.java

import p1.*;
public class item1
{
  public static void main(String args[])
  {
    item m=new item();
    m.read();
    m.display();
  }
}
```

OUTPUT

```
root@osboxes:~# mkdir p1
root@osboxes:~# cd p1
root@osboxes:~/p1# nano item.java
root@osboxes:~/p1# javac item.java
root@osboxes:~/p1# nano item.java
root@osboxes:~/p1# javac item.java
root@osboxes:~/p1# cd
root@osboxes:~# nano item1.java
root@osboxes:~# javac item1.java
root@osboxes:~# java item1
enter the item name
laptop
enter the item id
100
enter the item quantity
43
enter the item price
50000
Item name       Item id         Item quantity   Item price
laptop          100             43              50000
root@osboxes:~#
```

6.B) AVERAGE OF NUMBERS

```
//FILENAME average.java

package p2;
import java.util.*;
public class average
{
  int n,avg,s=0,i,no;
  Scanner sc=new Scanner(System.in);
  public void read()
  {
    System.out.println("enter the limit");
    n=sc.nextInt();
    System.out.println("enter the nos");
    int a[]=new int[n];
    for(i=0;i<n;i++)
    {
      a[i]=sc.nextInt();
    }
    for(i=0;i<n;i++)
    {
      s=s+a[i];
    }
    avg=s/n;
    System.out.println("average="+avg);
  }
}


//FILE NAME a

import p2.*;
public class a
{
  public static void main(String args[])
  {
    average av=new average();
    av.read();
```

```
}}
```

OUTPUT

```
root@osboxes:~# mkdir p2
root@osboxes:~# cd p2
root@osboxes:~/p2# nano average.java
root@osboxes:~/p2# javac average.java
root@osboxes:~/p2# cd
root@osboxes:~# nano a.java
root@osboxes:~# javac a.java
root@osboxes:~# java a
enter the limit
6
enter the nos
3 6 4 9 7 6
average=5
```

7. EXCEPTION HANDLING

```
import java.io.*;
import java.util.*;
class ex
{
 public static void main(String args[])
 {
  try
    {
     int a[]=new int[5];
     int l,s=0;
     Scanner sc= new Scanner(System.in);
     System.out.println("Enter the limit");
     l=sc.nextInt();
     System.out.println("Enter" +l+ "numbers");
     for(int i=0;i<l;i++)
      {
       a[i]=sc.nextInt();
       s=s+a[i];
      }
     System.out.println("Sum of numbers:"+s);
   }

 catch(ArrayIndexOutOfBoundsException e)
 {
  System.out.println(e);
 }
 catch(NegativeArraySizeException e)
 {
  System.out.println(e);
 }
 catch(ArithmeticException e)
 {
  System.out.println(e);
 }
 catch(NumberFormatException e)
 {
  System.out.println(e);
```

```
    }
  }
}
```

OUTPUT

```
root@student-Veriton-S2670G:~# nano ex.java
root@student-Veriton-S2670G:~# javac ex.java
root@student-Veriton-S2670G:~# java ex
Enter the limit
5
Enter5numbers
2
3
4
5
6
Sum of numbers:20
root@student-Veriton-S2670G:~#
```

8.A) EXTEND THREAD

```
class a extends Thread
{
   public void run()
   {
      for(int i=0;i<5;i++)
      {
         System.out.println(getName()+"="+i);
      }
   }
   a()
   {
      start();
   }
}
class testthreadclass
{
   public static void main(String args[])
   {
      a ob1=new a();
      a ob2=new a();
   }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano testthreadclass.java
root@osboxes:/home/osboxes# javac testthreadclass.java
root@osboxes:/home/osboxes# java testthreadclass
Thread-0=0
Thread-0=1
Thread-0=2
Thread-0=3
Thread-0=4
Thread-1=0
Thread-1=1
Thread-1=2
Thread-1=3
Thread-1=4
root@osboxes:/home/osboxes#
```

8.B) THREAD SYNCHRONIZATION

```
class table
{
   synchronized void printtable(int n)
   {
      for(int i=1;i<5;i++)
      {
         System.out.println(i+"*"+n+"="+n*i);
      }
   }
}
class mythread1 extends Thread
{
   table t;
   mythread1(table t)
   {
      this.t=t;
   }
   public void run()
   {
      synchronized(t)
      {
         t.printtable(100);
      }
   }
}
class mythread2 extends Thread
{
   table t;
   mythread2(table t)
   {
      this.t=t;
   }
   public void run()
   {
      t.printtable(5);
   }
}
```

8.B) THREAD SYNCHRONIZATION

```
class table
{
  synchronized void printtable(int n)
  {
    for(int i=1;i<5;i++)
    {
      System.out.println(i+"*"+n+"="+n*i);
    }
  }
}
class mythread1 extends Thread
{
  table t;
  mythread1(table t)
  {
    this.t=t;
  }
  public void run()
  {
    synchronized(t)
    {
      t.printtable(100);
    }
  }
}
class mythread2 extends Thread
{
  table t;
  mythread2(table t)
  {
    this.t=t;
  }
  public void run()
  {
    t.printtable(5);
  }
}
```

```
class threadsynchronization
{
   public static void main(String args[])
   {
      table ob=new table();
      mythread1 t1=new mythread1(ob);
      mythread2 t2=new mythread2(ob);
      t1.start();
      t2.start();
   }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano threadsynchronization.java
root@osboxes:/home/osboxes# javac threadsynchronization.java
root@osboxes:/home/osboxes# java threadsynchronization
1*100=100
2*100=200
3*100=300
4*100=400
1*5=5
2*5=10
3*5=15
4*5=20
root@osboxes:/home/osboxes#
```

8.C) THREAD SLEEP

```
import java.io.*;
class slp extends Thread
{
  public void run()
  {
    for(int i=1;i<5;i++)
    {
      try
      {
        Thread.sleep(500);
      }
      catch(InterruptedException e)
      {
        System.out.println(e);
      }
      System.out.println(i);
    }
  }
  public static void main(String args[])
  {
    slp t1=new slp();
    slp t2=new slp();
    t1.start();
    t2.start();
  }
}
```

OUTPUT

```
root@osboxes:/home/osboxes# nano slp.java
root@osboxes:/home/osboxes# javac slp.java
root@osboxes:/home/osboxes# java slp
1
1
2
2
3
3
4
4
root@osboxes:/home/osboxes#
```

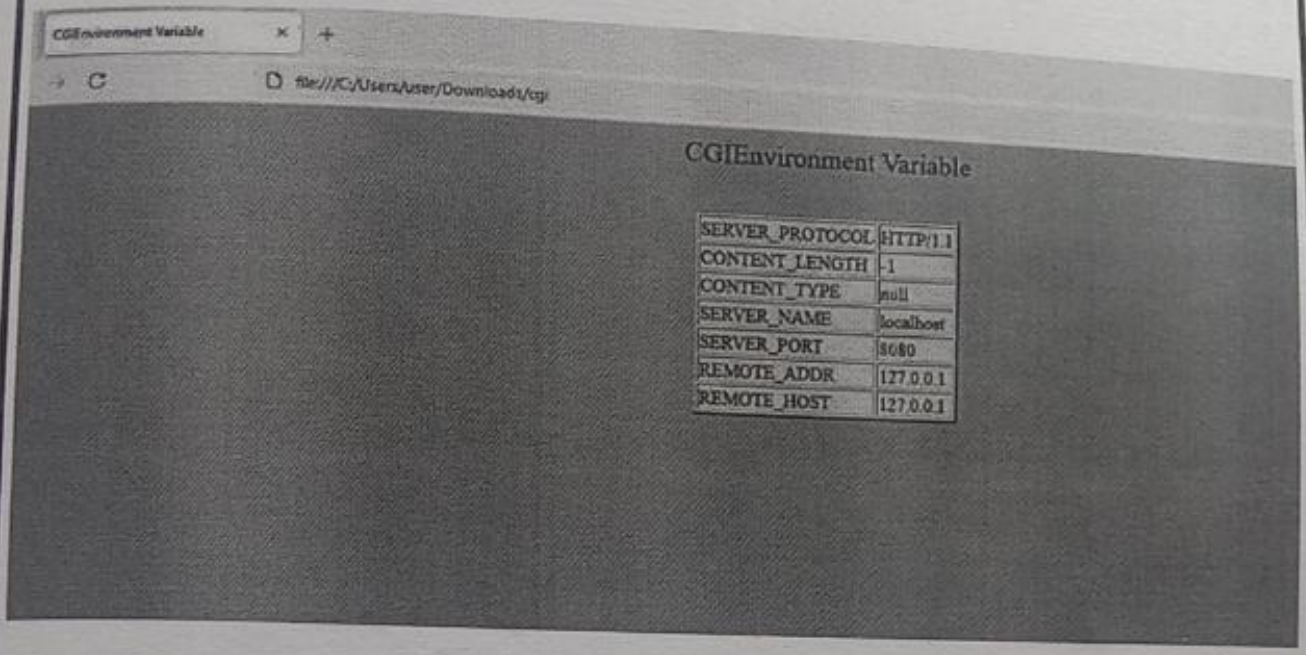
9.SERVLET PROGRAM TO PRINT THE ENVIRONMENTAL VARIABLES

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class cgi extends HttpServlet
{
   public void doGet(HttpServletRequest req,HttpServletResponse res)throws
   IOException,ServletException
   {
      res.setContentType("text/thtml");
      PrintWriter out=res.getWriter();
      out.println("<html><title>CGIEnvironment Variable</title><body bgcolor=violet><center><font
size=5 color=#000033>CGIEnvironment Variable</font></center>");
      out.println("<br><br><center><table bgcolor=#FFFF33 border=2>");
      out.println("<tr><td>SERVER_PROTOCOL</td><td>"+req.getProtocol()+"</td></tr>");
      out.println("<tr><td>CONTENT_LENGTH</td><td>"+req.getContentLength()+"</td></tr>");
      out.println("<tr><td>CONTENT_TYPE</td><td>"+req.getContentType()+"</td></tr>");
      out.println("<tr><td>SERVER_NAME</td><td>"+req.getServerName()+"</td></tr>");
      out.println("<tr><td>SERVER_PORT</td><td>"+req.getServerPort()+"</td></tr>");
      out.println("<tr><td>REMOTE_ADDR</td><td>"+req.getRemoteAddr()+"</td></tr>");
      out.println("<tr><td>REMOTE_HOST</td><td>"+req.getRemoteHost()+"</td></tr>");
      out.println("</table></body></html>");
   }
}
```

xml file

```
<webapp>
<servlet>
<servlet-name>cgi</servlet-name>
<servlet-class>cgi</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>cgi</servlet-name>
<url-pattern>/cgi</url-pattern>
</servlet-mapping>
</webapp>
```

OUTPUT


CGIEnvironment Variable
file:///C:/Users/user/Downloads/cgi
CGIEnvironment Variable
SERVER_PROTOCOL HTTP/1.1
CONTENT_LENGTH -1
CONTENT_TYPE null
SERVER_NAME localhost
SERVER_PORT 8080
REMOTE_ADDR 127.0.0.1
REMOTE_HOST 127.0.0.1

10 . Program to implement Arithmetic operation using RMI.

Calculator.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface Calculator extends Remote {
    int add(int a, int b) throws RemoteException;
    int subtract(int a, int b) throws RemoteException;
    int multiply(int a, int b) throws RemoteException;
    int divide(int a, int b) throws RemoteException;
}
```

CalcImpl.java

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class CalcImpl extends UnicastRemoteObject implements Calculator {
    public CalcImpl() throws RemoteException {
        super();
    }

    @Override
    public int add(int a, int b) throws RemoteException {
        return a + b;
    }

    @Override
    public int subtract(int a, int b) throws RemoteException {
        return a - b;
    }

    @Override
    public int multiply(int a, int b) throws RemoteException {

     return a * b;
    }

    @Override
    public int divide(int a, int b) throws RemoteException {
        if (b == 0) {
            throw new RemoteException("Cannot divide by zero");
        }
        return a / b;
    }
}
```

CalcServer.java

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class CalcServer {
   public static void main(String[] args) {
      try {
         Calculator calc = new CalcImpl();
         Registry registry = LocateRegistry.createRegistry(1099);
         registry.rebind("CalculatorService", calc);
         System.out.println("Calculator server is running.");
      } catch (Exception e) {
         System.err.println("Calculator server exception: " + e.toString());
         e.printStackTrace();
      }
   }
}
```

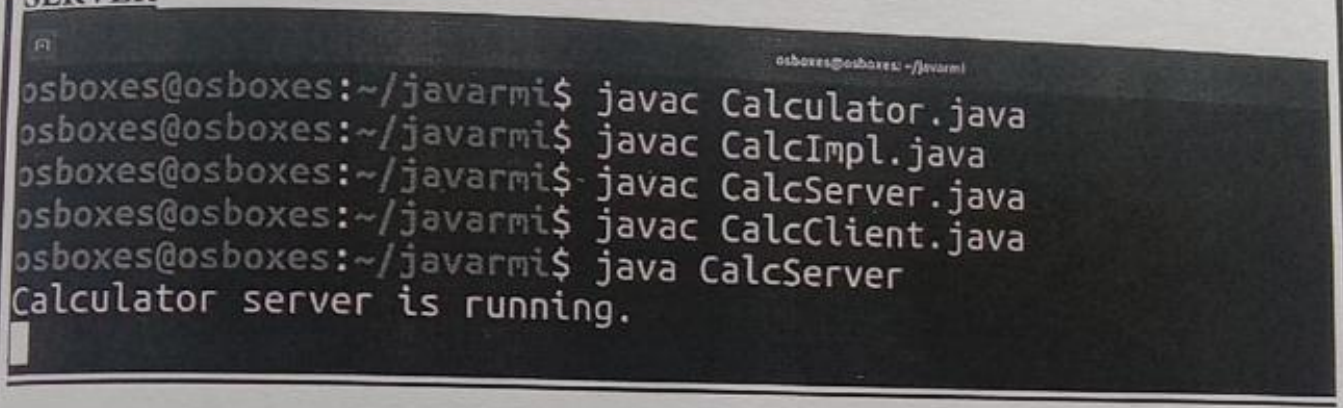
CalcClient.java

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class CalcClient {
   public static void main(String[] args) {
      try {
         Registry registry = LocateRegistry.getRegistry("localhost", 1099);
         Calculator calc = (Calculator) registry.lookup("CalculatorService");

         int a = 10;
         int b = 5;

         int sum = calc.add(a, b);
         System.out.println("Sum: " + sum);

         int difference = calc.subtract(a, b);
         System.out.println("Difference: " + difference);

         int product = calc.multiply(a, b);
         System.out.println("Product: " + product);

         int quotient = calc.divide(a, b);
         System.out.println("Quotient: " + quotient);
```

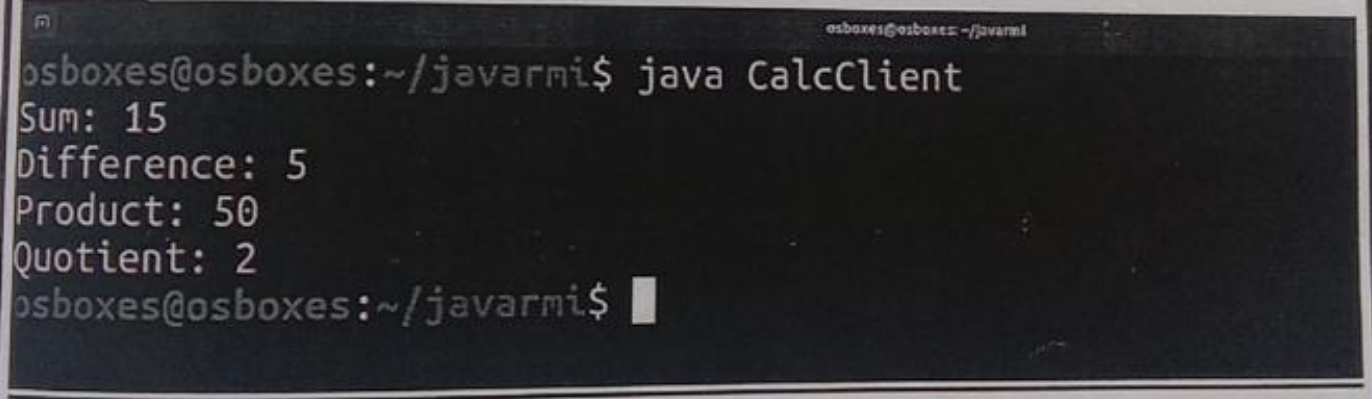
```
        } catch (Exception e) {
            System.err.println("Calculator client exception: " + e.toString());
            e.printStackTrace();
        }
    }
}
```

OUTPUT

SERVER



CLIENT



11. Implement client server communication using RMI

Hello.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface Hello extends Remote {
   String sayHello() throws RemoteException;
}
```

HelloImpl.java

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class HelloImpl extends UnicastRemoteObject implements Hello {
   public HelloImpl() throws RemoteException {
      super();
   }

   @Override
   public String sayHello() throws RemoteException {
      return "Hello, RMI!";
   }
}
```

HelloServer.java

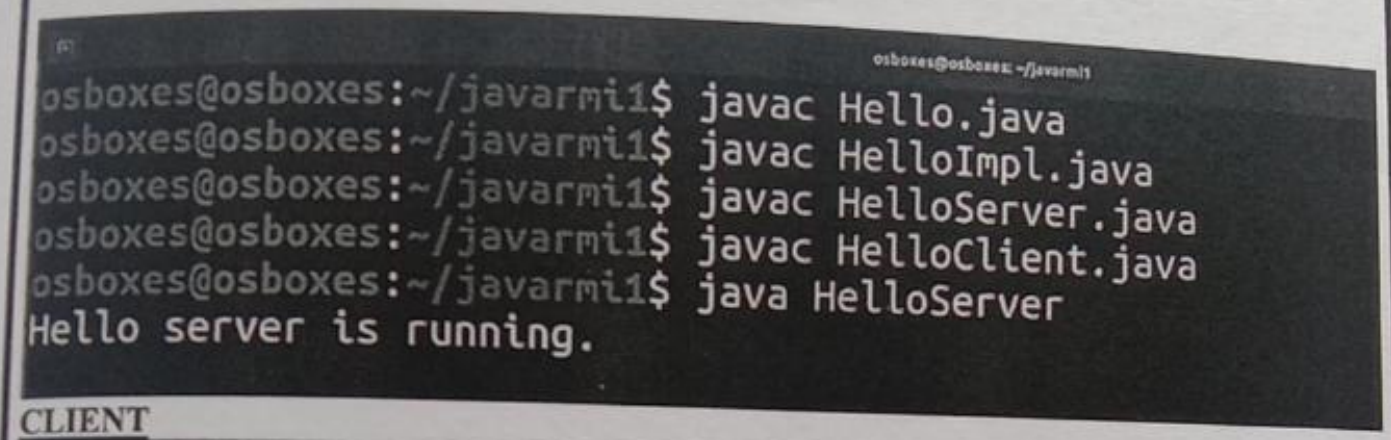
```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class HelloServer {
   public static void main(String[] args) {
      try {
         Hello hello = new HelloImpl();
         Registry registry = LocateRegistry.createRegistry(1099);
         registry.rebind("HelloService", hello);
         System.out.println("Hello server is running.");
      } catch (Exception e) {
         System.err.println("Hello server exception: " + e.toString());
         e.printStackTrace();
      }
   }
}
```

HelloClient.java

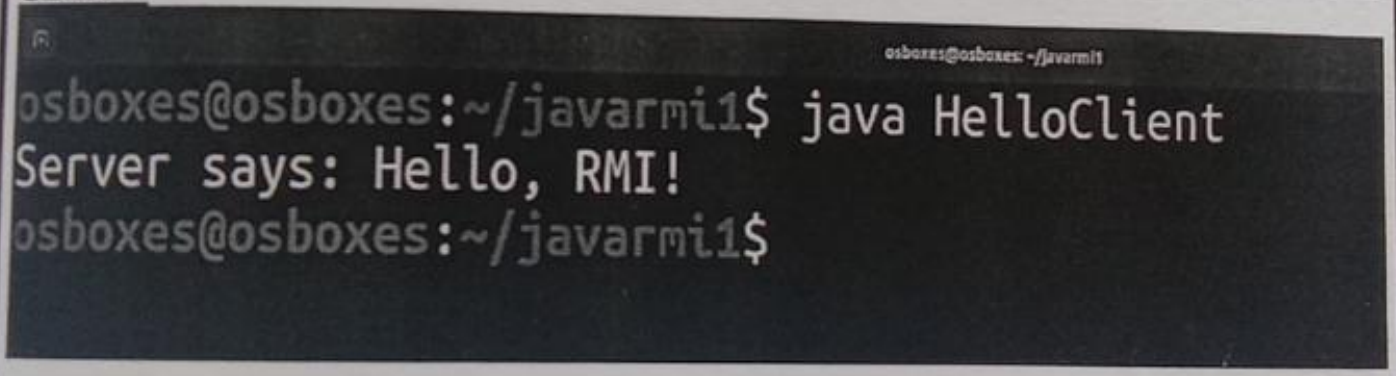
```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class HelloClient {
   public static void main(String[] args) {
      try {
         Registry registry = LocateRegistry.getRegistry("localhost", 1099);
         Hello hello = (Hello) registry.lookup("HelloService");

         String message = hello.sayHello();
         System.out.println("Server says: " + message);

      } catch (Exception e) {
         System.err.println("Hello client exception: " + e.toString());
         e.printStackTrace();
      }
   }
}
```

OUTPUT

SERVER



CLIENT



12. Implement Client Server Chat Application using RMI

ChatServer.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface ChatServer extends Remote {
    void registerClient(ChatClient client) throws RemoteException;
    void broadcastMessage(String message) throws RemoteException;
}
```

ChatClient.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface ChatClient extends Remote {
    void receiveMessage(String message) throws RemoteException;
}
```

ChatServerImpl.java

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import java.util.ArrayList;
import java.util.List;

public class ChatServerImpl extends UnicastRemoteObject implements ChatServer {
    private List<ChatClient> clients;

    public ChatServerImpl() throws RemoteException {
        clients = new ArrayList<>();
    }

    public void registerClient(ChatClient client) throws RemoteException {
        clients.add(client);
        System.out.println("Client registered.");
    }


public void broadcastMessage(String message) throws RemoteException {
    System.out.println("Broadcasting: " + message);
    for (ChatClient client : clients) {
        client.receiveMessage(message);
    }
```

```
    }
}
```

ChatClientImpl.java

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class ChatClientImpl extends UnicastRemoteObject implements ChatClient {
    private ChatServer server;

    public ChatClientImpl(ChatServer server) throws RemoteException {
        this.server = server;
        server.registerClient(this);
    }

    public void sendMessage(String message) throws RemoteException {
        server.broadcastMessage(message);
    }

    public void receiveMessage(String message) throws RemoteException {
        System.out.println("Received: " + message);
    }
}
```

ChatServerApp.java

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class ChatServerApp {
    public static void main(String[] args) {
        try {
            ChatServer server = new ChatServerImpl();
            Registry registry = LocateRegistry.createRegistry(1099);
            registry.rebind("ChatServer", server);
            System.out.println("Chat server is running.");

        } catch (Exception e) {
            System.err.println("Chat server exception: " + e.toString());
            e.printStackTrace();
        }
    }
}
```

ChatClientApp.java

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class ChatClientApp {
  public static void main(String[] args) {
    try {
      Registry registry = LocateRegistry.getRegistry("localhost", 1099);
      ChatServer server = (ChatServer) registry.lookup("ChatServer");

      ChatClientImpl client = new ChatClientImpl(server);
      client.sendMessage("Hello from the client!");

    } catch (Exception e) {
      System.err.println("Chat client exception: " + e.toString());
      e.printStackTrace();
    }
  }
}
```

OUTPUT

SERVER

```
osboxes@osboxes:~/javarni1$ javac ChatServer.java
osboxes@osboxes:~/javarni1$ javac ChatServerImpl.java
osboxes@osboxes:~/javarni1$ javac ChatServerApp.java
osboxes@osboxes:~/javarni1$ javac ChatClient.java
osboxes@osboxes:~/javarni1$ javac ChatClientImpl.java
osboxes@osboxes:~/javarni1$ javac ChatClientApp.java
osboxes@osboxes:~/javarni1$ java ChatServerApp
Chat server is running.
Client registered.
Broadcasting: Hello from the client!
```

CLIENT

```
osboxes@osboxes:~/javarni1$ javac ChatServer.java
osboxes@osboxes:~/javarni1$ javac ChatServerImpl.java
osboxes@osboxes:~/javarni1$ javac ChatServerApp.java
osboxes@osboxes:~/javarni1$ javac ChatClient.java
osboxes@osboxes:~/javarni1$ javac ChatClientImpl.java
osboxes@osboxes:~/javarni1$ javac ChatClientApp.java
osboxes@osboxes:~/javarni1$ java ChatClientApp
Received: Hello from the client!
```