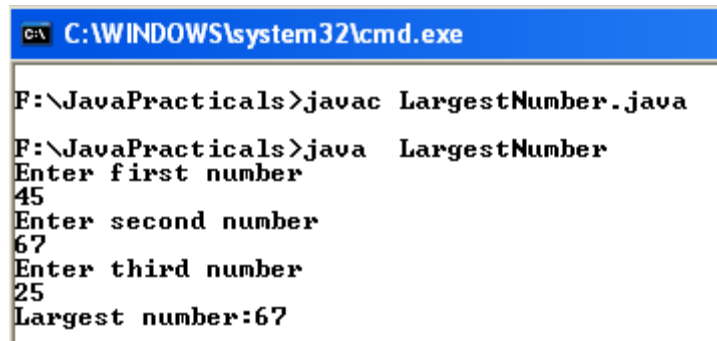


B.Sc (Computer Science)
Programming in Java
Lab Programs

- 1. Write java programs to find the following.**
a) Largest of given Three Numbers b) Reverses the digits of a number
c) Given number is prime or not d) GCD of given two integers.

a)

```
import java.util.Scanner;
class LargestNumber
{
    public static void main(String[] args)
    {
        int a,b,c;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter first number");
        a=s.nextInt();
        System.out.println("Enter second number");
        b=s.nextInt();
        System.out.println("Enter third number");
        c=s.nextInt();
        if(a>=b && a>=c)
            System.out.println("Largest number:" + a);
        else if (b>=a && b>=c)
            System.out.println("Largest number:" + b);
        else
            System.out.println("Largest number:" + c);
    }
}
```



```
C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac LargestNumber.java

F:\JavaPracticals>java LargestNumber
Enter first number
45
Enter second number
67
Enter third number
25
Largest number:67
```

b)

```
import java.util.Scanner;
class ReverseNumber
{
    public static void main(String[] args)
    {
        int n, rev=0;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter first number");
        n=s.nextInt();
        while(n!=0)
```

```

    {
        int d=n%10;
        rev=rev*10+d;
        n=n/10;
    }
    System.out.println("Reversed Number:" + rev);
}
}

```

```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac ReverseNumber.java

F:\JavaPracticals>java ReverseNumber
Enter first number
456
Reversed Number:654

```

c)

```

import java.util.Scanner;
class PrimeNumber
{
    public static void main(String[] args)
    {
        int n, count=0;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the number");
        n=s.nextInt();
        for (int i=1;i<=n ;i++ )
        {
            if(n%i ==0)
                count++;
        }
        if(count>2)
            System.out.println("Give number is Not Prime");
        else
            System.out.println("Given number is Prime");
    }
}

```

```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac PrimeNumber.java

F:\JavaPracticals>java PrimeNumber
Enter the number
17
Given number is Prime

F:\JavaPracticals>_

```

d)

```

import java.util.Scanner;
class GCD
{
    public static void main(String[] args)

```

```

    {
        int n1, n2;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the first number");
        n1=s.nextInt();
        System.out.println("Enter the second number");
        n2=s.nextInt();
        int res=gcd(n1,n2);
        System.out.println("G.C.D of" + n1 + "and " + n2 + "is " + res);
    }

    static int gcd(int n1, int n2)
    {
        if (n2 != 0)
            return gcd(n2, n1%n2);
        else
            return n1;
    }
}

```

```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac GCD.java
F:\JavaPracticals>java GCD
Enter the first number
48
Enter the second number
36
G.C.D of 48 and 36 is 12
F:\JavaPracticals>

```

2. Write java programs that implement the following
a) default constructor b) parameterized constructor c) constructor overloading

a)

```

import java.util.Scanner;

class Rectangle
{
    double width;
    double height;

    Rectangle()
    {
        width=10;
        height=20;
    }

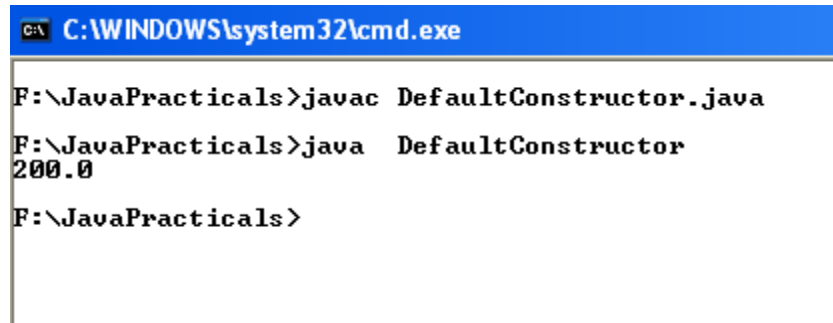
    double area()
    {
        return width*height;
    }
}

```

```

class DefaultConstructor
{
    public static void main(String[] args)
    {
        Rectangle r1=new Rectangle();
        System.out.println(r1.area());
    }
}

```



```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac DefaultConstructor.java
F:\JavaPracticals>java DefaultConstructor
200.0
F:\JavaPracticals>

```

b)

```
import java.util.Scanner;
```

```

class Rectangle
{
    double width;
    double height;

    Rectangle(double w,double h)
    {
        width=w;
        height=h;
    }

    double area()
    {
        return width*height;
    }
}

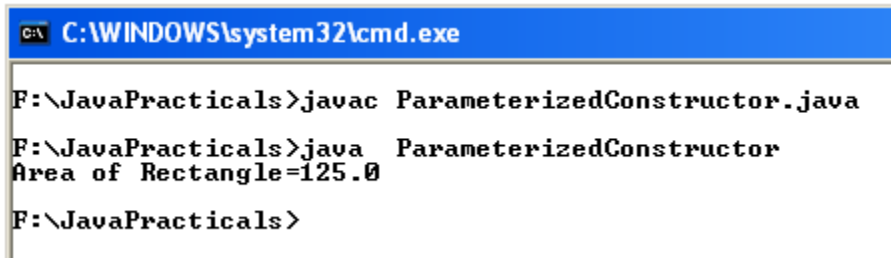
```

```
class ParameterizedConstructor
```

```

{
    public static void main(String[] args)
    {
        Rectangle r2=new Rectangle(5,25);
        System.out.println("Area of Rectangle=" + r2.area());
    }
}

```



```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac ParameterizedConstructor.java
F:\JavaPracticals>java ParameterizedConstructor
Area of Rectangle=125.0
F:\JavaPracticals>

```

c)

```

import java.util.Scanner;

class Rectangle
{
    double width;
    double height;

    Rectangle()
    {
        width=10;
        height=20;
    }

    Rectangle(double w,double h)
    {
        width=w;
        height=h;
    }

    double area()
    {
        return width*height;
    }
}

class ConstructorOverloading
{
    public static void main(String[] args)
    {
        Rectangle r1=new Rectangle();
        System.out.println("Area of First Rectangle=" + r1.area());
        Rectangle r2=new Rectangle(5,25);
        System.out.println("Area of Second Rectangle=" + r2.area());
    }
}

```

```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac ConstructorOverloading.java

F:\JavaPracticals>java ConstructorOverloading
Area of First Rectangle=200.0
Area of Second Rectangle=125.0

F:\JavaPracticals>

```

3. (a). Write a java program to find the smallest of given list integers using array & scanner class.
(b) . Write a java program for multiplication of two matrices.

a)

```

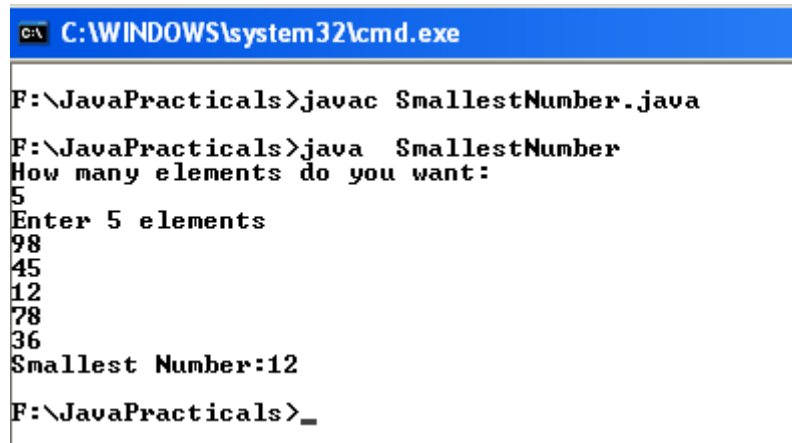
import java.util.Scanner;
class SmallestNumber
{
    public static void main(String[] args)

```

```

    {
        Scanner s=new Scanner(System.in);
        System.out.println("How many elements do you want:");
        int n=s.nextInt();
        int a[]=new int[n];
        System.out.println("Enter " + n + " elements");
        for(int i=0;i<n;i++)
            a[i]=s.nextInt();
        int min=a[0];
        for(int i=1;i<n;i++)
            if(a[i]<min)
                min=a[i];
        System.out.println("Smallest Number:" + min);
    }
}

```



```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac SmallestNumber.java

F:\JavaPracticals>java SmallestNumber
How many elements do you want:
5
Enter 5 elements
98
45
12
78
36
Smallest Number:12

F:\JavaPracticals>_

```

b)

```

import java.util.Scanner;
class MatrixMultiplication
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);

        int m1[][]=new int[3][3];
        int m2[][]=new int[3][3];
        int m3[][]=new int[3][3];
        int i,j,k;
        System.out.println("Enter elements of first matrix (3 x 3 Matrix) ");
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                m1[i][j]=s.nextInt();
            }
        }

        System.out.println("Enter elements of second matrix (3 x 3 Matrix) ");
        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                m2[i][j]=s.nextInt();
            }
        }
    }
}

```

```

/*Matrix multiplication*/
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
m3[i][j]=0;
for(k=0;k<3;k++)
{
//m3[i][j]+=m1[i][k]*m2[k][j];
m3[i][j]=m3[i][j]+m1[i][k]*m2[k][j];
}
}
}
}

```

```

System.out.println("Matrix Multiplication result is ");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
System.out.print(m3[i][j] + " ");
}
System.out.println();
}
}
}
}

```

```

C:\WINDOWS\system32\cmd.exe
F:\JavaPracticals>javac MatrixMultiplication.java
F:\JavaPracticals>java MatrixMultiplication
Enter elements of first matrix (3 x 3 Matrix)
7 8 9
0 0 0
0 0 0
Enter elements of second matrix (3 x 3 Matrix)
1 0 0
0 1 0
0 0 1
Matrix Multiplication result is
7 8 9
0 0 0
0 0 0
F:\JavaPracticals>_

```

4. a) Write a java program for demonstrating an inner classes or nested classes.
 b) Write a java program to implement method overloading & method overriding

a)

```

class Outer
{
void test()
{
Inner i=new Inner();
i.display();
}
}

class Inner
{

```

```

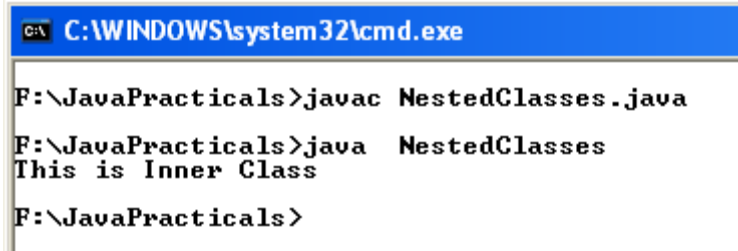
    void display()
    {
        System.out.println("This is Inner Class");
    }
}
}

```

```

class NestedClasses
{
    public static void main(String args[])
    {
        Outer o=new Outer();
        o.test();
    }
}

```



```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac NestedClasses.java
F:\JavaPracticals>java NestedClasses
This is Inner Class
F:\JavaPracticals>

```

b)

Method Overloading:

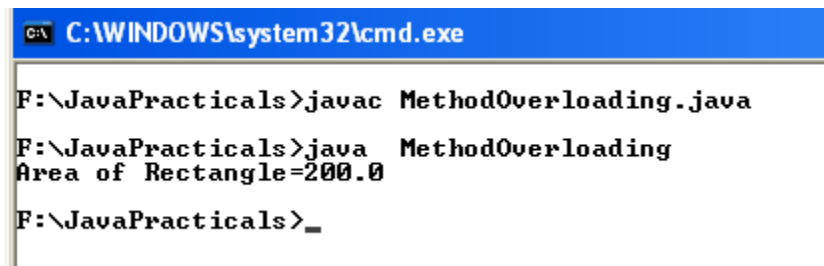
```

class Rect
{
    double width;
    double height;

    double area(double w,double h)
    {
        width=w;
        height=h;
        return width*height;
    }
}

class MethodOverloading
{
    public static void main(String[] args)
    {
        Rect r1=new Rect();
        System.out.println("Area of Rectangle=" + r1.area(10,20));
    }
}

```



```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac MethodOverloading.java
F:\JavaPracticals>java MethodOverloading
Area of Rectangle=200.0
F:\JavaPracticals>_

```


Method Overriding:

```

class A
{
int i, j;
A(int a, int b) {
i = a;
j = b;
}
void show() {
System.out.println("i and j: " + i + " " + j);
}
}

```

```

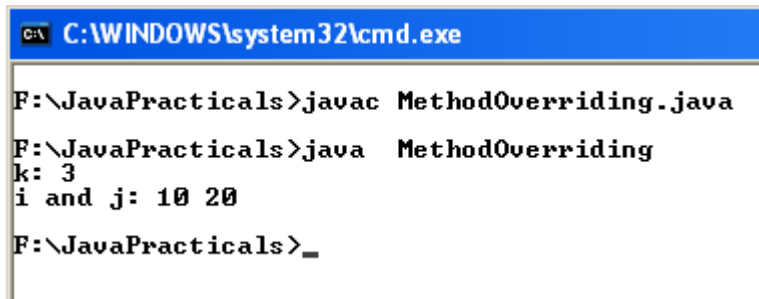
class B extends A {
int k;
B(int a, int b, int c) {
super(a, b);
k = c;
}
void show() {
System.out.println("k: " + k);
}
}

```

```

class OverrideMethod {
public static void main(String args[]) {
B b = new B(1, 2, 3);
b.show();
A a=new A(10,20);
a.show();
}
}

```



```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac MethodOverriding.java
F:\JavaPracticals>java MethodOverriding
k: 3
i and j: 10 20
F:\JavaPracticals>_

```

- 7. a) Write a java program for creating a package and using a package.
b) Write a java program to demonstrate the use of wrapper classes.**

a)

```

package Arithmetic;

public class mathsop
{
    public int add(int a,int b)
    {
        return a+b;
    }
    public int sub(int a,int b)
    {
        return a-b;
    }
}

```

```

    }
public int mult(int a,int b)
    {
        return a*b;
    }
public double div(int a,int b)
    {
        return a/b;
    }
}

```

Note: Store the above file in Arithmetic folder.

```

import Arithmetic.mathsop;
import java.util.Scanner;

```

```

public class PackageExample
{
    public static void main(String[] args)
    {
        System.out.println("Enter the values of A and B:");
        Scanner s=new Scanner(System.in);
        int a=s.nextInt();
        int b=s.nextInt();
        mathsop m=new mathsop();
        System.out.println("Addition=" + m.add(a,b) + "\nSubtraction=" +
m.sub(a,b) + "\nMultiplication=" + m.mult(a,b)+ "\nDivision=" + m.div(a,b));

    }
}

```

Note: Store this file outside Arithmetic folder.

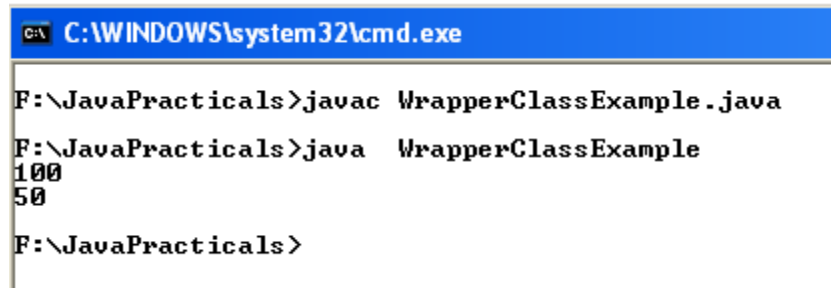
```

C:\WINDOWS\system32\cmd.exe
F:\JavaPracticals\Arithmetic>javac mathsop.java
F:\JavaPracticals\Arithmetic>cd..
F:\JavaPracticals>javac PackageExample.java
F:\JavaPracticals>java PackageExample
Enter the values of A and B:
10
20
Addition=30
Subtraction=-10
Multiplication=200
Division=0.0
F:\JavaPracticals>

```

b)

```
public class WrapperClassExample
{
    public static void main(String[] args)
    {
        int a=100;
        Integer i=Integer.valueOf(a); // converting int into Integer object
        System.out.println(i);
        int j=Integer.parseInt("50"); // converting String object to int
        System.out.println(j);
    }
}
```



```
C:\WINDOWS\system32\cmd.exe
F:\JavaPracticals>javac WrapperClassExample.java
F:\JavaPracticals>java WrapperClassExample
100
50
F:\JavaPracticals>
```

- 8. a) Write a java program using Exception handling mechanism.
b) Write a java program for creating customized (user) exception**

a)

```
import java.util.Scanner;

class ExceptionExample
{
    public static void main(String as[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter A and B");
        int a=s.nextInt();
        int b=s.nextInt();
        try
        {
            System.out.println("Division="+ (a/b));
        }
        catch(ArithmeticException e)
        {
            System.out.println("Division by zero is not possible");
        }
    }
}
```

```

C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>java ExceptionExample
Enter A and B
20
10
Division=2

F:\JavaPracticals>java ExceptionExample
Enter A and B
20
0
Division by zero is not possible

F:\JavaPracticals>

```

b)

```

import java.util.Scanner;
class UserDefinedException
{
    public static void main(String[] args)
    {
        try
        {
            //User defined exception if marks>100

            Scanner s=new Scanner(System.in);
            System.out.println("Enter the marks in Java:");
            int mk=s.nextInt();
            if (mk>100)
            {
                throw new MarksOutOfBound("Marks should not be greater than
100.");
            }
            System.out.println("Entered marks: " + mk+ " is valid");
        }
        catch (MarksOutOfBound mb)
        {
            System.out.println(mb.getMessage());
        }
    }
}

class MarksOutOfBound extends Exception
{
    MarksOutOfBound(String s)
    {
        super(s);
    }
}

```

```
C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>java UserDefinedException
Enter the marks in Java:
100
Entered marks: 100 is valid

F:\JavaPracticals>java UserDefinedException
Enter the marks in Java:
101
Marks should not be greater than 100.

F:\JavaPracticals>_
```

9. a) Write a java program that checks whether a given string is a palindrome or not.
b) Write a java program for sorting a given list of names in ascending order.

a)

```
import java.util.Scanner;
class StringPalindrome
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the String:");
        String str=s.next();
        System.out.println("String is:" + str);
        String reversestring="";
        for (int i = str.length() - 1; i >= 0; i--)
            reversestring = reversestring + str.charAt(i);
        System.out.println("String is:" + reversestring);
        if (str.equals(reversestring))
            System.out.println("The string is a palindrome.");
        else
            System.out.println("The string isn't a palindrome.");
    }
}
```

```
C:\WINDOWS\system32\cmd.exe

F:\JavaPracticals>javac StringPalindrome.java
F:\JavaPracticals>java StringPalindrome
Enter the String:
madam
String is:madam
String is:madam
The string is a palindrome.

F:\JavaPracticals>java StringPalindrome
Enter the String:
computer
String is:computer
String is:retupmoc
The string isn't a palindrome.

F:\JavaPracticals>_
```

b)

```

import java.util.*;

class StringAscending
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter number of strings:");
        int n=s.nextInt();
        System.out.println("Enter " + n + " String:");
        String name[]=new String[n];
        for(int i=0;i<n;i++)
            name[i]=s.next();
        System.out.println("-----");

        String temp=null;
        for(int i=0;i<name.length;i++)
        {
            for(int j=i+1;j<name.length;j++)
            {
                if(name[j].compareTo(name[i])<0)
                {
                    temp=name[i];
                    name[i]=name[j];
                    name[j]=temp;
                }
            }
        }

        System.out.println("Sorted Strings:");
        System.out.println("-----");

        for(int i=0;i<name.length;i++)
        {
            System.out.println(name[i]);
        }
    }
}

```

```

C:\WINDOWS\system32\cmd.exe

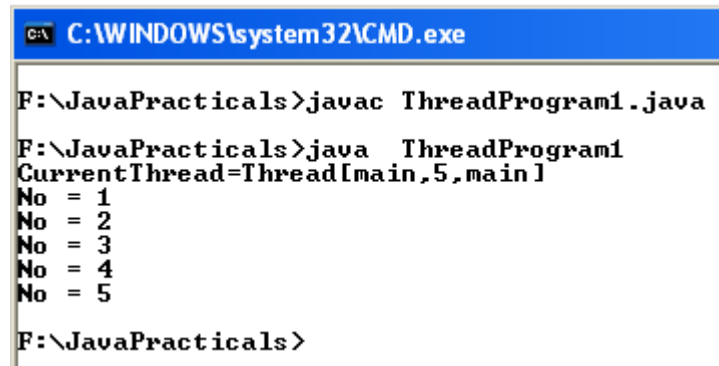
F:\JavaPracticals>javac StringAscending.java
F:\JavaPracticals>java StringAscending
Enter number of strings:
4
Enter 4 String:
java
c
unix
c++
-----
Sorted Strings:
-----
c
c++
java
unix
F:\JavaPracticals>

```

- 11. a) Write a java program for controlling main thread.
b) Write a java program for creating new thread by extending Thread class.**

a)

```
class ThreadProgram1
{
public static void main(String args[])
{
Thread t=Thread.currentThread();
System.out.println("CurrentThread=" + t);
for(int i=1;i<=5;i++)
{
System.out.println("No = " +i);
}
}
}
}
```



```
C:\WINDOWS\system32\CMD.exe

F:\JavaPracticals>javac ThreadProgram1.java

F:\JavaPracticals>java ThreadProgram1
CurrentThread=Thread[main,5,main]
No = 1
No = 2
No = 3
No = 4
No = 5

F:\JavaPracticals>
```

b)

```
class A extends Thread
{
public void run()
{
System.out.println("Thread A Started");
for(int i=1;i<=5;i++)
{
System.out.println("Thread A -> No = " +i);
}
System.out.println("Exit from Thread A ");
}
}

class B extends Thread
{
public void run()
{
System.out.println("Thread B Started");
for(int i=1;i<=5;i++)
{
System.out.println("Thread B -> No = " +i);
}
System.out.println("Exit from Thread B ");
}
}
}
```

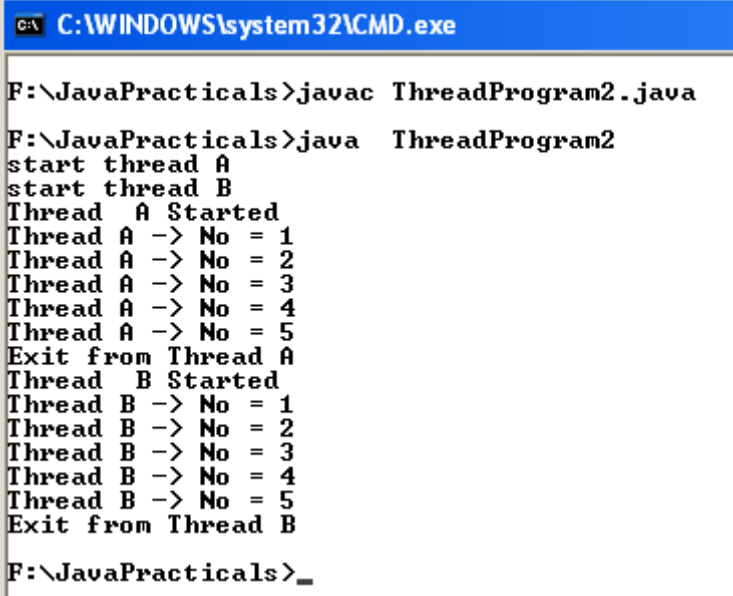
```

class ThreadProgram2
{
public static void main(String args[])
{
    A obja=new A();
    B objb=new B();

System.out.println("start thread A");
obja.start();
System.out.println("start thread B");
objb.start();

}
}

```



```

C:\WINDOWS\system32\CMD.exe

F:\JavaPracticals>javac ThreadProgram2.java

F:\JavaPracticals>java ThreadProgram2
start thread A
start thread B
Thread A Started
Thread A -> No = 1
Thread A -> No = 2
Thread A -> No = 3
Thread A -> No = 4
Thread A -> No = 5
Exit from Thread A
Thread B Started
Thread B -> No = 1
Thread B -> No = 2
Thread B -> No = 3
Thread B -> No = 4
Thread B -> No = 5
Exit from Thread B

F:\JavaPracticals>_

```

- 12. a) Write a java program for creating new thread by implementing Runnable interface.
b) Write a java program for thread synchronization.**

a)

```

public class ThreadProgram3 implements Runnable
{
    public void run()
    {
        System.out.println("Hello from a thread!");
    }

    public static void main(String args[])
    {
        ThreadProgram3 tp3=new ThreadProgram3();
        tp3.run();
    }
}

```



```
C:\WINDOWS\system32\CMD.exe
F:\JavaPracticals>javac ThreadProgram3.java
F:\JavaPracticals>java ThreadProgram3
Hello from a thread!
F:\JavaPracticals>
```

b)

```
import java.util.Scanner;

class SyncProgram
{
    void printValues(int n)
    {
        synchronized(this){//synchronized block
        for(int i=1;i<=5;i++)
        {
            System.out.println(i);
        }
    }
}
class A extends Thread
{
    SyncProgram sp;
    A(SyncProgram sp)
    {
        this.sp=sp;
    }

    public void run()
    {
        sp.printValues(5);
    }
}
class B extends Thread
{
    SyncProgram sp;
    B(SyncProgram sp)
    {
        this.sp=sp;
    }

    public void run()
    {
        sp.printValues(5);
    }
}

class ThreadProgram4
{
    public static void main(String args[])
    {
        SyncProgram obj=new SyncProgram();
```

```

A t1=new A(obj);
B t2=new B(obj);

System.out.println("start thread A");
t1.start();
System.out.println("start thread B");
t2.start();

}
}

```

```

C:\WINDOWS\system32\CMD.exe

F:\JavaPracticals>javac ThreadProgram4.java
F:\JavaPracticals>java ThreadProgram4
start thread A
start thread B
1
2
3
4
5
1
2
3
4
5
F:\JavaPracticals>_

```

14. Write java programs to create a simple Applet and create swing based Applet.

Applet File: (SimpleApplet.java)

```

import java.awt.*;
import java.applet.*;
public class SimpleApplet extends Applet
{
public void paint (Graphics g)
{
g.drawString("Hello java", 10, 100);
}
}

```

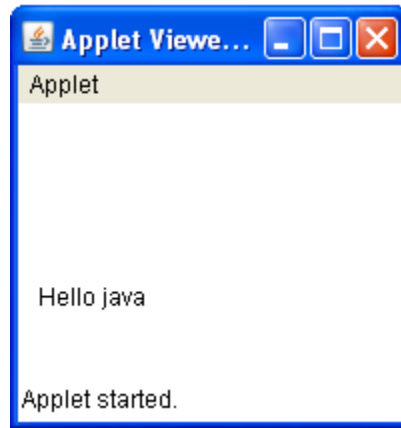
HTML File: (AppletExample.java)

```

<HTML>
<HEAD>
<TITLE>
Welcome to java applet
</TITLE>
</HEAD>
<BODY>
<APPLET CODE = "SimpleApplet.class"
        WIDTH = 400
        HEIGHT = 200>
</APPLET>
</BODY>
</HTML>

```

```
C:\WINDOWS\system32\cmd.exe - appletviewer AppletExample.html
F:\JavaPracticals>javac SimpleApplet.java
F:\JavaPracticals>appletviewer AppletExample.html
```

Result:**Swing File:**

```
import javax.swing.JApplet;
import java.awt.*;
public class SwingApplet extends JApplet
{
public void paint (Graphics g)
{
g.drawOval(25,25,100,200); // (x,y, width,height)
}
}
```

HTML File:

```
<HTML>
<HEAD>
<TITLE>
Welcome to java applet
</TITLE>
</HEAD>
<BODY>
<APPLET CODE = "SwingApplet.class"
        WIDTH = 400
        HEIGHT = 200>
</APPLET>
</BODY>
</HTML>
```

```
C:\WINDOWS\system32\cmd.exe - appletviewer SwingExample.html
F:\JavaPracticals>javac SwingApplet.java
F:\JavaPracticals>appletviewer SwingExample.html
```

Result: