

**:: INDEX ::**

<b>Serial No.</b>	<b>Program</b>	<b>Teacher Signature</b>
<b>1</b>	<b>To find the sum of any number of integers entered as command line arguments</b>	
<b>2</b>	<b>To find the factorial of a given number</b>	
<b>3</b>	<b>To learn use of single dimensional array by defining the array dynamically</b>	
<b>4</b>	<b>To learn use of .length in case of a two dimensional array</b>	
<b>5</b>	<b>To convert a decimal to binary number</b>	
<b>6</b>	<b>To check if a number is prime or not, by taking the number as input from the keyboard</b>	
<b>7</b>	<b>To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument</b>	
<b>8</b>	<b>Write a program that show working of different functions of String and StringBuffer classes like setCharAt(), setLength(), append(), insert(), concat() and equals()</b>	
<b>9</b>	<b>Write a program to create a —distance class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer</b>	
<b>10</b>	<b>Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions (from lower to higher data type)</b>	
<b>11</b>	<b>Example of Single Inheritance</b>	
<b>12</b>	<b>Example of Multi Level Inheritance</b>	

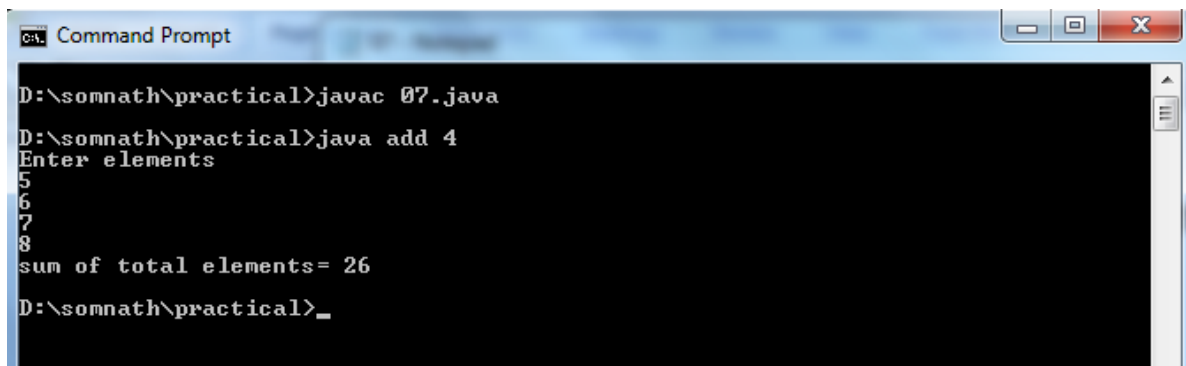
<b>13</b>	<b>Example of Hierarchical Inheritance</b>	
<b>14</b>	<b>Write a program to demonstrate the concept of boxing and unboxing</b>	
<b>15</b>	<b>Example Of Method Overloading</b>	
<b>16</b>	<b>Example of Method Overriding</b>	
<b>17</b>	<b>Example of Package in JAVA</b>	
<b>18</b>	<b>Write a program —DivideByZero that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero</b>	
<b>19</b>	<b>Example of Interface in JAVA</b>	
<b>20</b>	<b>Example of Applet in JAVA</b>	
<b>21</b>	<b>Example of Thread in JAVA</b>	

- To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument

#### PROGRAM CODE:

```
import java.util.Scanner;
class add
{
    public static void main(String args[])
    {
        int n=Integer.parseInt(args[0]);
        int a[]= new int[n];
        int sum=0;
        Scanner sc=new Scanner (System.in);
        System.out.println("Enter elements");
        for(int i=0;i<n;i++)
        {
            a[i]=sc.nextInt();
        }
        for(int i=0;i<n;i++)
        {
            sum=sum+a[i];
        }
        System.out.println("sum of total elements= "+sum);
    }
}
```

#### OUTPUT:



```
Command Prompt
D:\somnath\practical>javac 07.java
D:\somnath\practical>java add 4
Enter elements
5
6
7
8
sum of total elements= 26
D:\somnath\practical>_
```

TEACHER SIGNATURE:

- **Write a program that show working of different functions of String and StringBuffer classes like setCharAt(), setLength(), append(), insert(), concat() and equals()**

**PROGRAM CODE:**

```
import java.io.*;

class strings {

    public static void main (String args[]) {

        InputStreamReader read=new InputStreamReader(System.in);

        BufferedReader in=new BufferedReader(read);

        String str1="DRAGON";

        String str2="FLY";

        StringBuffer s1=new StringBuffer("DRAGON");

        StringBuffer s2=new StringBuffer("AGE");

        s1.setCharAt(2,'O');

        System.out.println("\nafter setCharAt() string is: "+s1);

        s1.setLength(4);

        System.out.println("\nnew string after change length: "+s1);

        s1.append(s2);

        System.out.println("\nAfter append: "+s1);

        s1.insert(5," TO ");

        System.out.println("\nAfter insert(): "+s1);
```

```
String s3=str1.concat(str2);

System.out.println("\nAfter concat(): "+s3+"\n\n");


String s4="FLY";

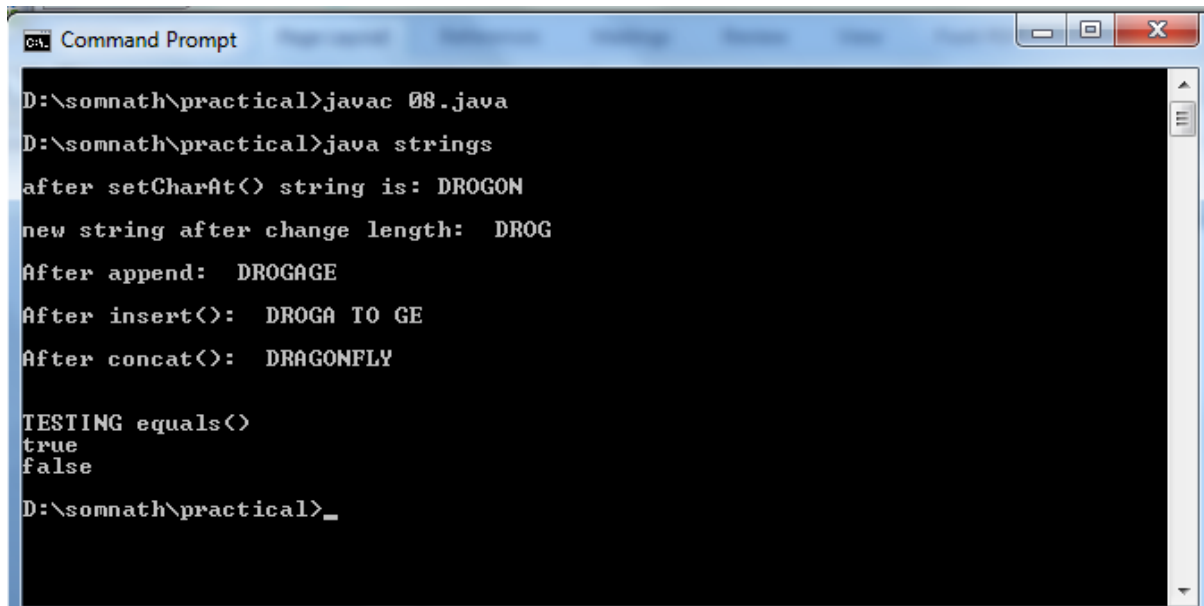
System.out.println("TESTING equals()");

System.out.println(s4.equals(str2));

System.out.println(str2.equals(s1));


}}
```

**OUTPUT:**



```
cmd - Command Prompt
D:\somnath\practical>javac 08.java
D:\somnath\practical>java strings
after setCharAt() string is: DROGON
new string after change length: DROG
After append: DROGAGE
After insert(): DROGA TO GE
After concat(): DRAGONFLY

TESTING equals()
true
false
D:\somnath\practical>
```

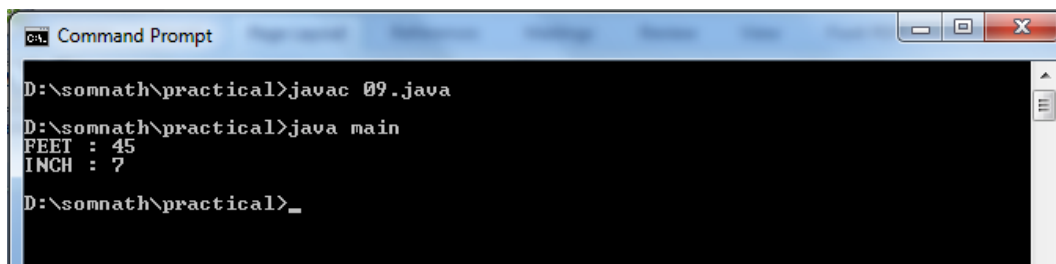
**TEACHER SIGNATURE:**

- Write a program to create a —distance class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer

**PROGRAM CODE:**

```
class distance {  
    int feet;  
    int inch;  
    void display()  
    {  
        System.out.println("FEET : "+feet);  
        System.out.println("INCH : "+inch);  
    }  
    void getdistance(int feet, int inch)  
    {  
        this.feet=feet;  
        this.inch=inch;  
    } }  
class main {  
    public static void main (String args[]) {  
        distance obj=new distance();  
        obj.getdistance(45,7);  
        obj.display();  
    } }
```

**OUTPUT:**



```
Command Prompt  
D:\somnath\practical>javac 09.java  
D:\somnath\practical>java main  
FEET : 45  
INCH : 7  
D:\somnath\practical>_
```

**TEACHER SIGNATURE:**

- **Example of Single Inheritance**

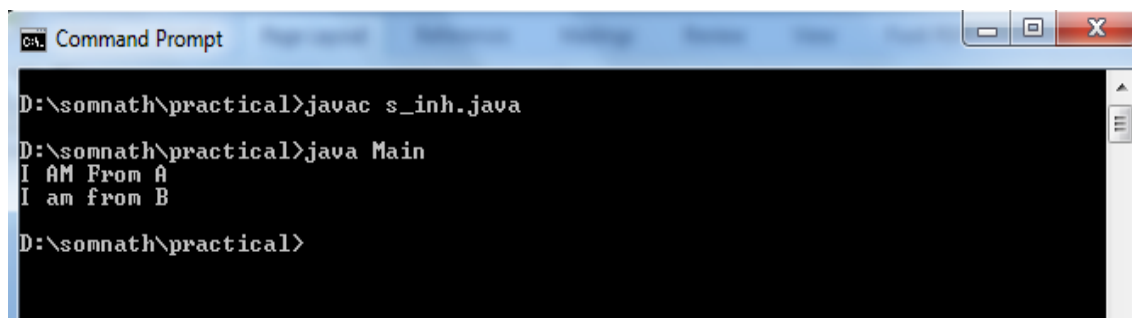
**PROGRAM CODE:**

```
class A
{
    public void displayA()
    {
        System.out.println("I AM From A");
    }
}

class B extends A
{
    public void displayB()
    {
        System.out.println("I am from B");
    }
}

class Main {
    public static void main(String args[]) {
        B obj=new B();
        obj.displayA();
        obj.displayB();
    }
}
```

**OUTPUT:**



```
Command Prompt
D:\somnath\practical>javac s_inh.java
D:\somnath\practical>java Main
I AM From A
I am from B
D:\somnath\practical>
```

**TEACHER SIGNATURE:**

- **Example of Multi Level Inheritance**

**PROGRAM CODE:**

```
class A    {

public void displayA()    {

System.out.println("I am from A");

}}

class B extends A    {

public void displayB()    {

System.out.println("I am from B");

}}

class C extends B    {

public void displayC()    {

System.out.println("I am from C");

}}

class Main_M {

public static void main(String args[]) {

C obj=new C();

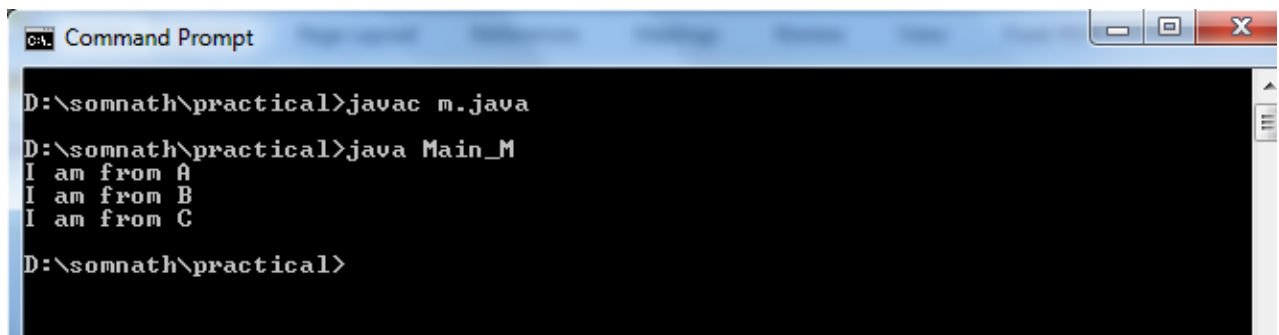
obj.displayA();

obj.displayB();

obj.displayC();

}}
```

**OUTPUT:**



```
Command Prompt
D:\somnath\practical>javac m.java
D:\somnath\practical>java Main_M
I am from A
I am from B
I am from C
D:\somnath\practical>
```

**TEACHER SIGNATURE:**

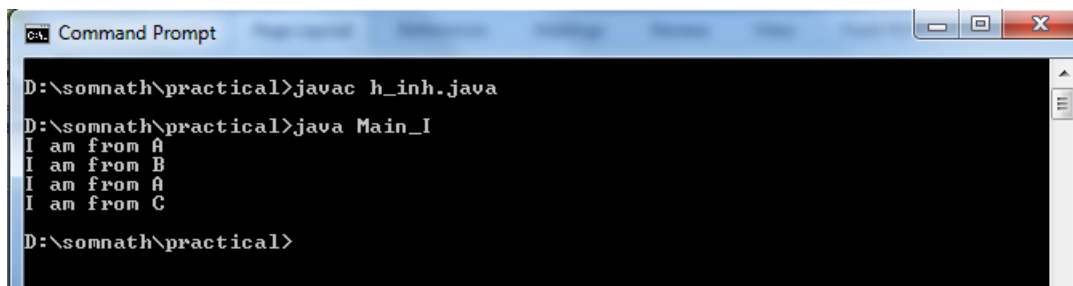


- **Example of Hierarchical Inheritance**

**PROGRAM CODE:**

```
class A    {  
    public void displayA()    {  
        System.out.println("I am from A");  
    } }  
  
class B extends A    {  
    public void displayB()    {  
        System.out.println("I am from B");  
    } }  
  
class C extends A    {  
    public void displayC()    {  
        System.out.println("I am from C");  
    } }  
  
class Main_I {  
    public static void main (String args[]) {  
        B obj1=new B();  
        C obj2=new C();  
        obj1.displayA();  
        obj1.displayB();  
        obj2.displayA();  
        obj2.displayC();  
    }  
}
```

**OUTPUT:**



```
Command Prompt  
D:\somnath\practical>javac h_inh.java  
D:\somnath\practical>java Main_I  
I am from A  
I am from B  
I am from A  
I am from C  
D:\somnath\practical>
```

**TEACHER SIGNATURE:**

- Write a program to demonstrate the concept of boxing and unboxing

**PROGRAM CODE:**

```
import java.util.Stack;

class b

{

public static void main (String args[])

{

Stack<Integer> myStack = new Stack<Integer>();

myStack.push(30);

myStack.push(40);

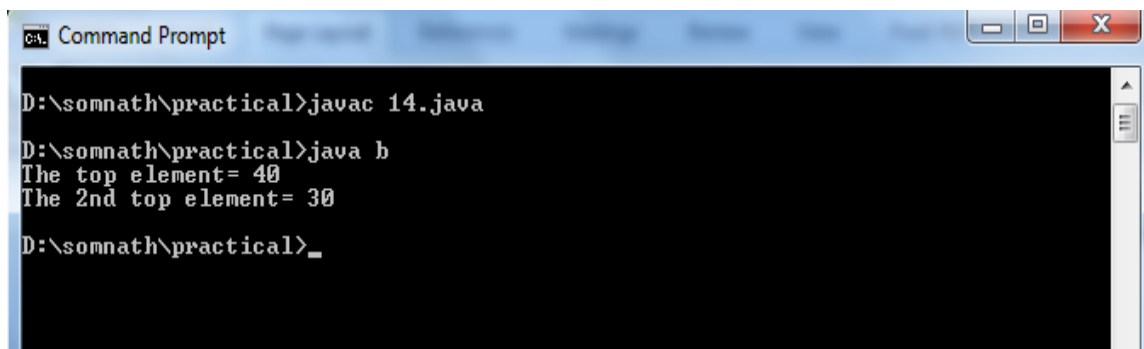
System.out.println("The top element= "+myStack.pop());

System.out.println("The 2nd top element= "+myStack.pop());

}

}
```

**OUTPUT:**



```
Command Prompt
D:\somnath\practical>javac 14.java
D:\somnath\practical>java b
The top element= 40
The 2nd top element= 30
D:\somnath\practical>_
```

**TEACHER SIGNATURE:**

- **Example Of Method Overloading**

**PROGRAM CODE:**

```
class add

{

void sum(int x, int y)

{

System.out.println("Sum= "+(x+y));

}

void sum(int x,int y,int z)

{

System.out.println("Sum of three number= "+(x+y+z));

}

}

class addM {

public static void main(String args[]) {

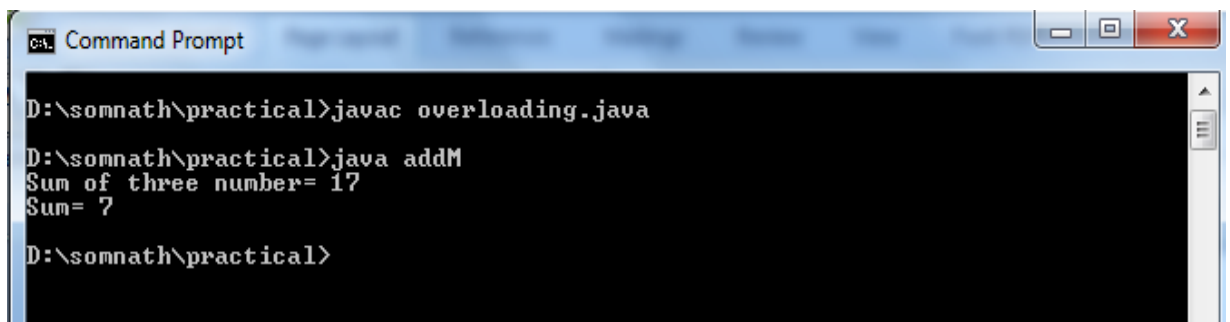
add obj=new add();

obj.sum(4,6,7);

obj.sum(3,4);

} }
```

**OUTPUT:**



```
Command Prompt

D:\somnath\practical>javac overloading.java

D:\somnath\practical>java addM
Sum of three number= 17
Sum= 7

D:\somnath\practical>
```

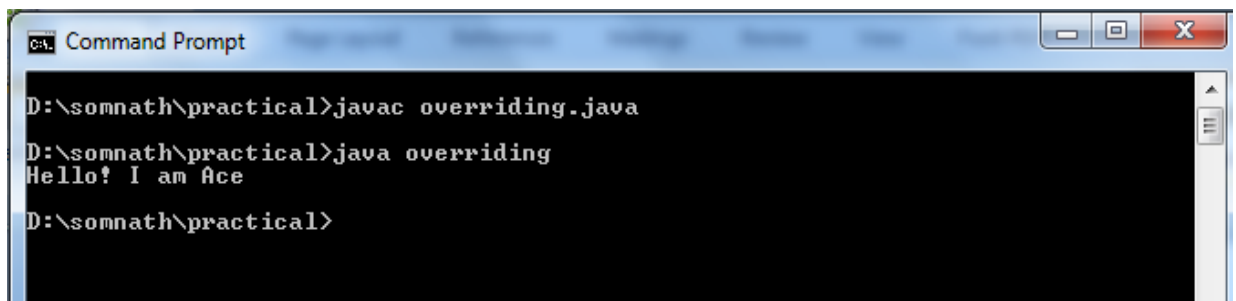
**TEACHER SIGNATURE:**

- **Method Overriding**

**PROGRAM CODE:**

```
class p {  
    void method()  
{  
    System.out.println("Hello! I am Game");  
}  
}  
  
class c extends p  
{  
    void method()  
{  
    System.out.println("Hello! I am Ace");  
} }  
  
class overriding {  
    public static void main(String args[]) {  
        c obj=new c();  
        obj.method();  
    } }
```

**OUTPUT:**



```
Command Prompt  
D:\somnath\practical>javac overriding.java  
D:\somnath\practical>java overriding  
Hello! I am Ace  
D:\somnath\practical>
```

**TEACHER SIGNATURE:**

- **Example of Package in JAVA**

**INSIDE OF PACKAGE:**

```
package add;  
  
public class sum {  
  
    public void sum(int x, int y) {  
  
        System.out.println("Sum= "+(x+y));  
  
    } }  

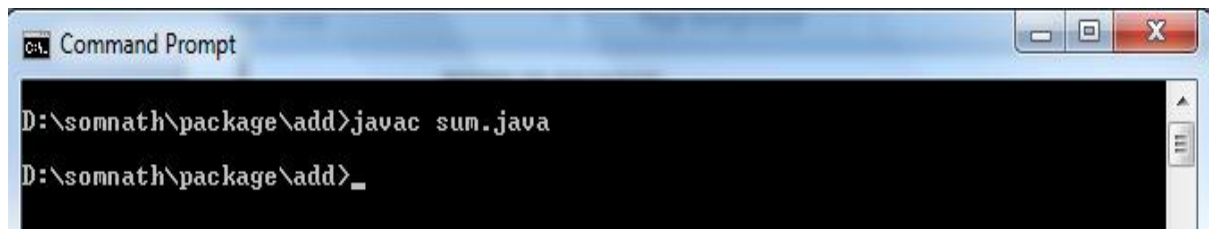
```

**PROGRAM CODE:**

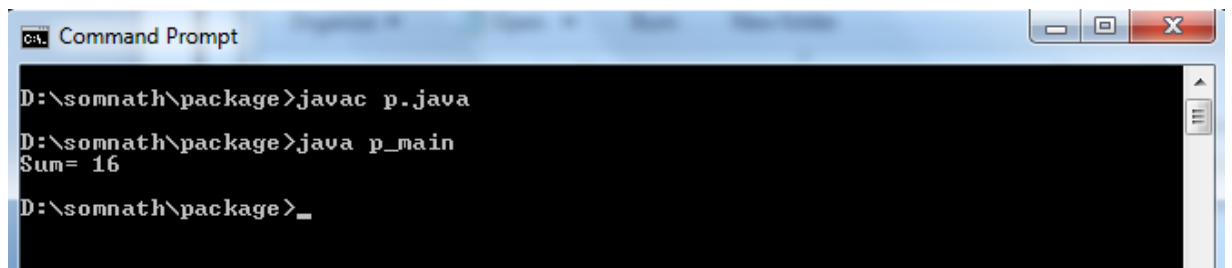
```
import add.sum;  
  
class p_main {  
  
    public static void main(String args[]) {  
  
        sum obj=new sum();  
  
        obj.sum(7,9);  
  
    } }  

```

**OUTPUT:**



A screenshot of a Windows Command Prompt window. The title bar reads "C:\> Command Prompt". The command prompt shows the directory "D:\somnath\package\add" and the command "javac sum.java" being executed. The prompt then shows "D:\somnath\package\add>\_" indicating the command was successful.



A screenshot of a Windows Command Prompt window. The title bar reads "C:\> Command Prompt". The command prompt shows the directory "D:\somnath\package" and the command "javac p.java" being executed. The prompt then shows "D:\somnath\package>java p\_main" being executed, followed by the output "Sum= 16". The prompt then shows "D:\somnath\package>\_" indicating the command was successful.

**TEACHER SIGNATURE:**

- **Example of Interface in JAVA**

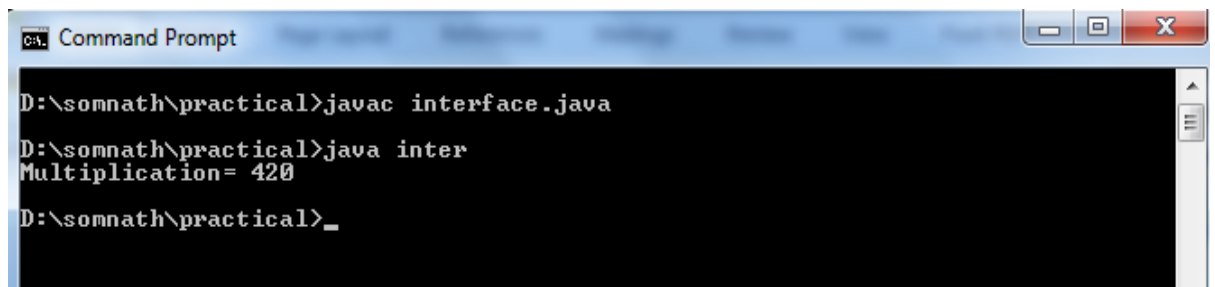
**PROGRAM CODE:**

```
interface inf
{
    final static int z=10;
    int mul(int x, int y);
}

class mul1 implements inf
{
    public int mul(int x, int y)
    {
        return(x*y*z);
    } }

class inter {
    public static void main (String args[]) {
        mul1 obj=new mul1();
        inf obj2;
        obj2=obj;
        int n=obj2.mul(6,7);
        System.out.println("Multiplication= "+n);
    } }
```

**OUTPUT:**



```
Command Prompt
D:\somnath\practical>javac interface.java
D:\somnath\practical>java inter
Multiplication= 420
D:\somnath\practical>_
```

**TEACHER SIGNATURE:**

- **Example of Applet in JAVA**

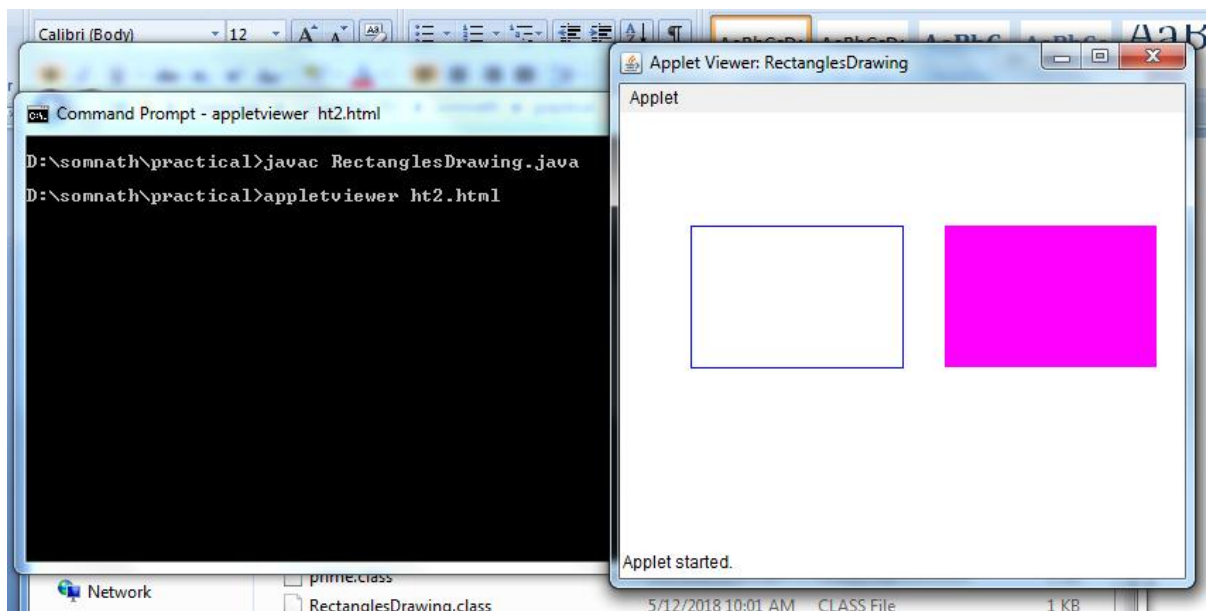
**PROGRAM CODE:**

```
import java.awt.*;  
import java.applet.*;  
public class RectanglesDrawing extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.setColor(Color.blue);  
        g.drawRect(50, 80, 150, 100);  
        g.setColor(Color.magenta);  
        g.fillRect(230, 80, 150, 100);  
    }  
}
```

**HTML CODE:**

```
<applet code="RectanglesDrawing" width="400" height="300">  
</applet>
```

**OUTPUT:**



**TEACHER SIGNATURE:**

- **Example of Thread in JAVA**

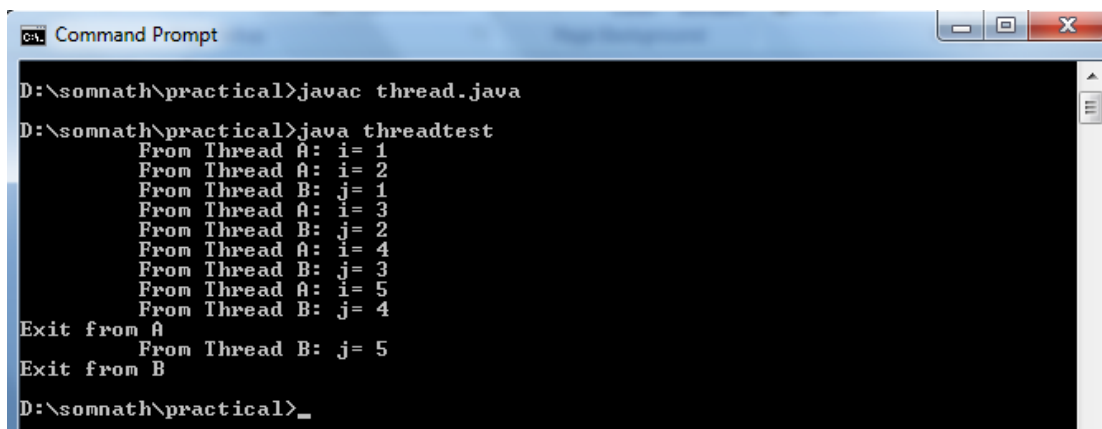
**PROGRAM CODE:**

```
class A extends Thread {
    public void run() {
        for(int i=1; i<=5; i++) {
            System.out.println("\t From Thread A: i= "+i);
        }
        System.out.println("Exit from A");
    } }

class B extends Thread {
    public void run() {
        for(int j=1; j<=5; j++) {
            System.out.println("\t From Thread B: j= "+j);
        }
        System.out.println("Exit from B");
    } }

class threadtest {
    public static void main(String args[]) {
        new A().start();
        new B().start();
    } }
```

**OUTPUT:**



```
Command Prompt
D:\somnath\practical>javac thread.java
D:\somnath\practical>java threadtest
    From Thread A: i= 1
    From Thread A: i= 2
    From Thread B: j= 1
    From Thread A: i= 3
    From Thread B: j= 2
    From Thread A: i= 4
    From Thread B: j= 3
    From Thread A: i= 5
    From Thread B: j= 4
Exit from A
    From Thread B: j= 5
Exit from B
D:\somnath\practical>_
```

**TEACHER SIGNATURE:**