

Index

S. No.	Practical	Page No.	Remarks
1	Introduction to Java	2-4	
2	To show grade of a student using IF statement	5	
3	Program to show series of '*' using while loop.	6	
4	Program to generate Fibonacci series using 'for' loop	7	
5	Program to Use of Do-while loop to show 10 numbers	8	
6	Program to determine which season a particular month is in	9	
7	Program to determine greatest number out of three numbers	10-11	
8	Program to Addition of two numbers and subtraction Algorithm	12	
9	Program to find a number is odd or even	13	
10	Program to find a number Prime or not	14	
11	Program to calculate the area of a circle, rectangle	15	
12	Program to find average of n numbers using arrays	16	
13	Program to find factorial of a number	17	
14	Program to find perimeter of a circle using final keyword	18	
15	Program which does the creation of class and object .Algorithm to find volume of a box	19	
16	Program to find volume of a box using constructor	20-21	
17	To find factorial of a number by passing reference of class factorial to the recursion class.	22	

Practical 1:- Introduction to JAVA

Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991. It took 18 months to develop the first working version. This language was initially called “Oak” but was renamed “Java” in 1995. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities.

Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA),



Illustration 1: Java Logo

Principles

There were five primary goals in the creation of the Java language:

1. It should be "simple, object-oriented and familiar"
2. It should be "robust and secure"
3. It should be "architecture-neutral and portable"
4. It should execute with "high performance"
5. It should be "interpreted, threaded, and dynamic"

Versions

Major release versions of Java, along with their release dates:

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)

Java platform

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to platform-specific machine code. Java bytecode instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets.

Features of Java

Object-Oriented

Although influenced by its predecessors, Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean, usable, pragmatic approach to objects. Borrowing liberally from many seminal object-software environments of the last few decades, Java manages to strike a balance between the purist's "everything is an object" paradigm and the pragmatist's "stay out of my way" model. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance nonobjects.

Robust

The multiplatformed environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. Thus, the ability to create robust programs was given a high priority in the design of Java.

Multithreaded

Java was designed to meet the real-world requirement of creating interactive, networked programs. To accomplish this, Java supports multithreaded programming, which allows you to write programs that do many things simultaneously.

Architecture-Neutral

A central issue for the Java designers was that of code longevity and portability. One of the main problems facing programmers is that no guarantee exists that if you write a program today, it will run tomorrow—even on the same machine. Operating system upgrades, processor upgrades, and changes in core system resources can all combine to make a program malfunction.

The Java designers made several hard decisions in the Java language and the Java Virtual Machine in an attempt to alter this situation. Their goal was "write once; run anywhere, any time, forever." To a great extent, this goal was accomplished.

Interpreted and High Performance

As described earlier, Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java bytecode. This code can be interpreted on any system that

provides a Java Virtual Machine. Most previous attempts at cross- platform solutions have done so at the expense of performance. Java run-time systems that provide this feature lose none of the benefits of the platform-independent code. “High-performance cross-platform” is no longer an oxymoron.

Distributed

Java is designed for the distributed environment of the Internet, because it handles TCP/IP protocols. In fact, accessing a resource using a URL is not much different from accessing a file. This feature brings an unparalleled level of abstraction to client/ server programming.

Dynamic

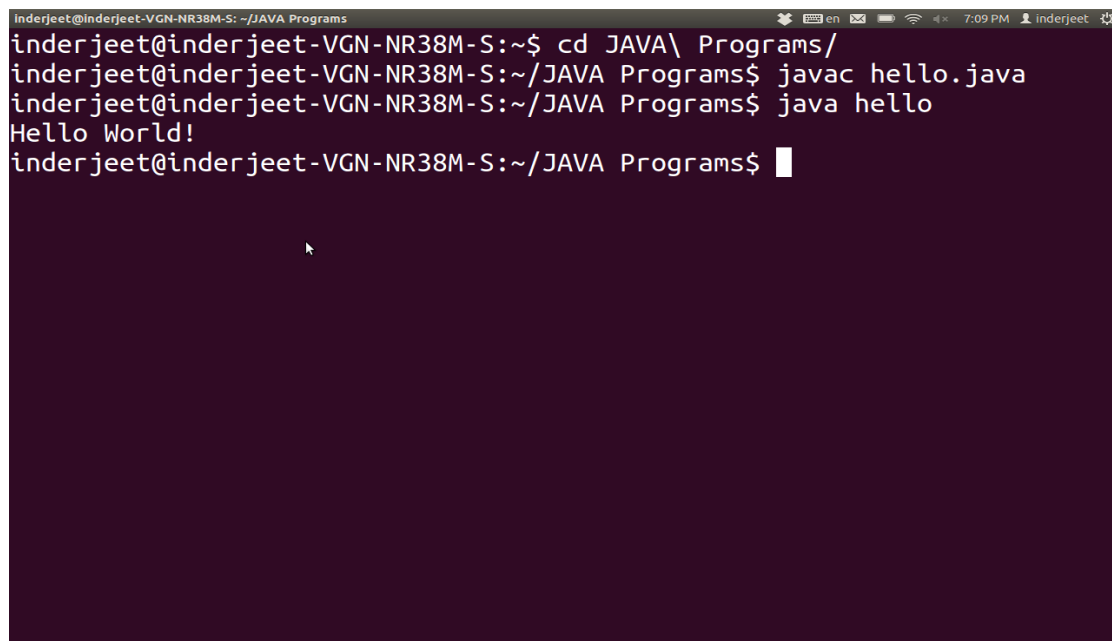
Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run time .

Basic structure of program

The traditional Hello world program can be written in Java as:-

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!"); // Display the string.  
    }  
}
```

Output of Program



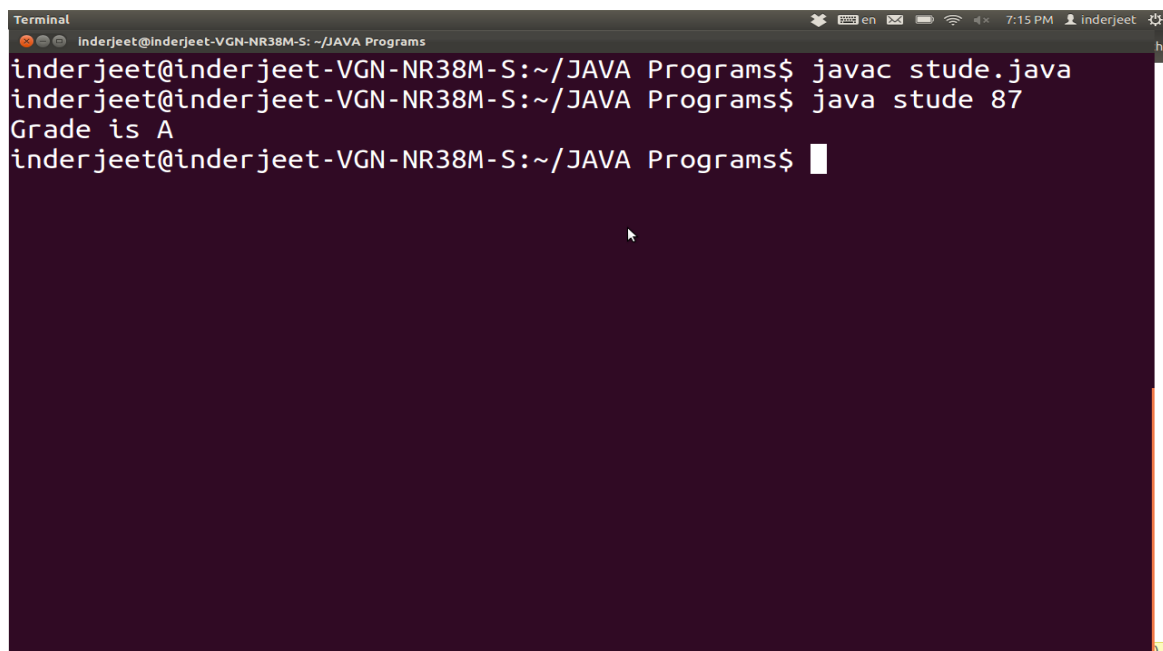
```
inderjeet@inderjeet-VGN-NR38M-S: ~/JAVA Programs  
inderjeet@inderjeet-VGN-NR38M-S:~$ cd JAVA\ Programs/  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$ javac hello.java  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$ java hello  
Hello World!  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$
```

Practical 2:- To show grade of a student using IF statement

Coding

```
class stude{
    public static void main(String args[]) {
        int c=60,b=40;
        int a= Integer.parseInt(args[0]);
        {
            if(a>c)
                System.out.println("Grade is A");
            else if(a<c&& a>b)
                System.out.println("Grade is B");
            else
                System.out.println("Grade is C");
        }
    }
}
```

Output

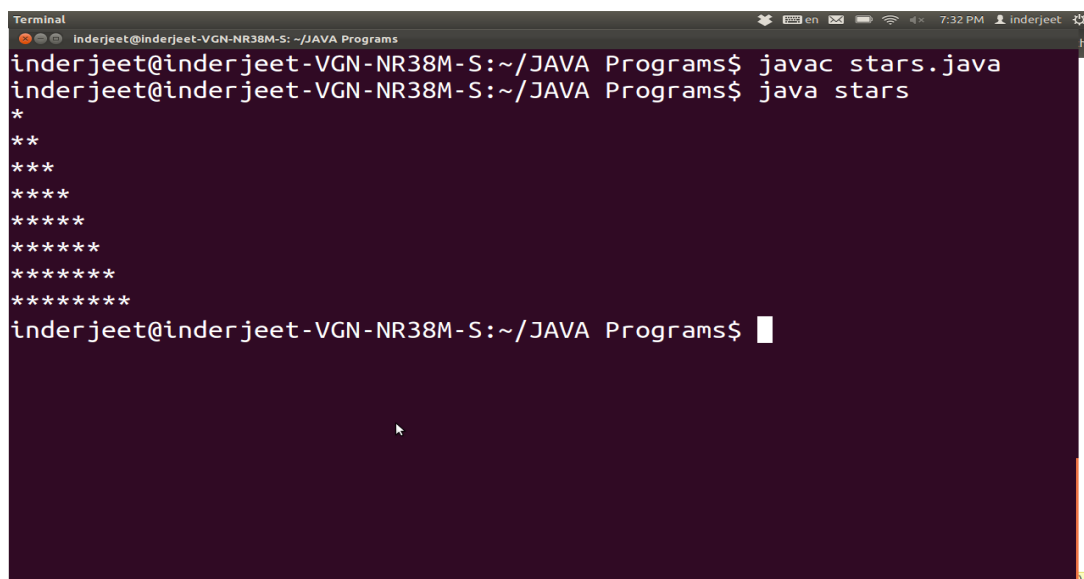
A screenshot of a terminal window with a dark background. The terminal shows the following commands and output:
1. Command: `javac stude.java`
2. Command: `java stude 87`
3. Output: `Grade is A`
The terminal title bar indicates the user is 'inderjeet' and the current directory is '~/JAVA Programs'. The system clock shows 7:15 PM.

Practical 3:- Program to show series of '*' using while loop.

Coding

```
class stars{  
    /** Below function holds logic to print triangle ****/  
    public static void printTriangle (int count) {  
        int line = 1;  
        while(line <= count) {  
            for(int x = 1; x <= line; x++) {  
                System.out.print("*");  
            }  
            System.out.print("\n");  
            line++;  
        }  
    }  
    /**Public function to call printTriangle****/  
    public static void main(String args[]){  
        printTriangle (8);  
    } }  
}
```

Output



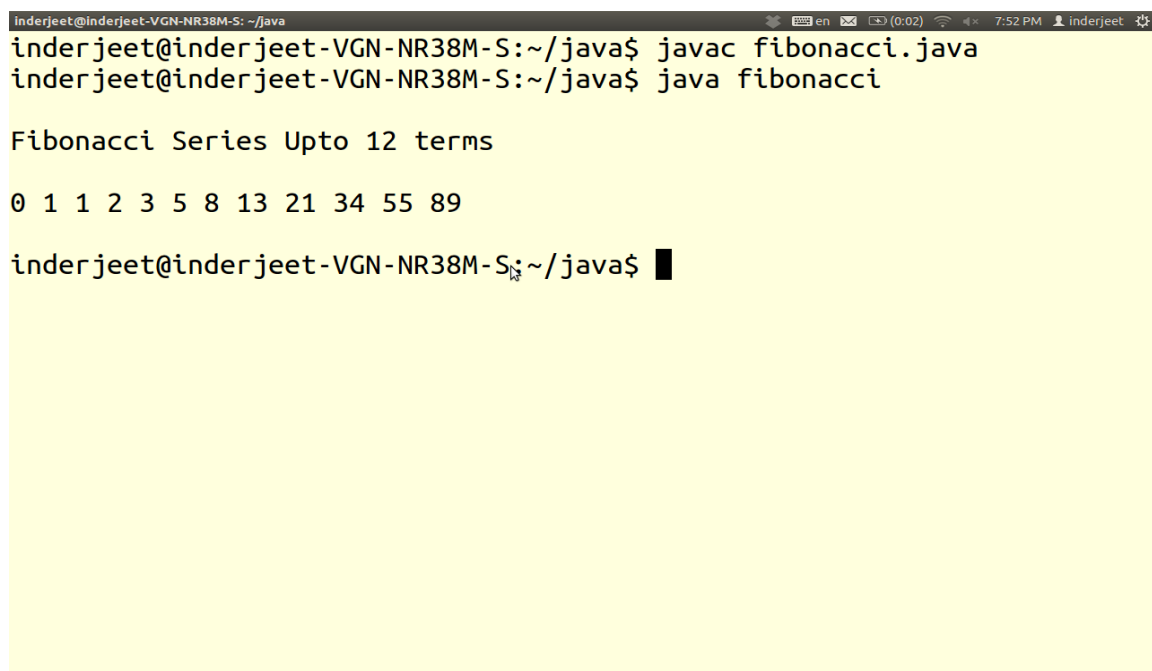
```
Terminal  
inderjeet@inderjeet-VGN-NR38M-S: ~/JAVA Programs  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$ javac stars.java  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$ java stars  
*  
**  
***  
****  
*****  
*****  
*****  
*****  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$
```

Practical 4:- Program to generate Fibonacci series using 'for' loop

Coding

```
class fibonacci{
    public static void main(String[] args) {
        int sum=0;
        int num =10;
        int f1=0,f2=1;
        System.out.println();
        System.out.println("Fibonacci Series Upto 12 terms ");
        System.out.println();
        System.out.print(f1+" "+f2+" ");
        for(int i=1;i<=num;i++){
            sum= f1+f2;
            f1=f2;
            f2=sum;
            System.out.print(sum+" ");
        }
        System.out.println();
        System.out.println();
    }
}
```

Output



```
inderjeet@inderjeet-VGN-NR38M-S: ~/java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ javac fibonacci.java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java fibonacci

Fibonacci Series Upto 12 terms

0 1 1 2 3 5 8 13 21 34 55 89

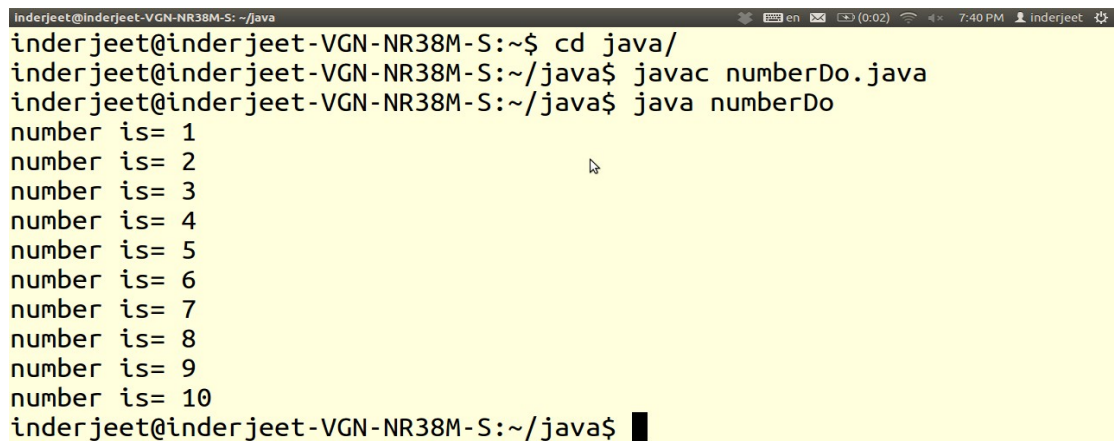
inderjeet@inderjeet-VGN-NR38M-S:~/java$
```

Practical 5:- Program to Use of Do-while loop to show 10 numbers

Coding

```
class numberDo
{
    public static void main(String args[]) {
        int i=1, n=10;
        do
        {
            System.out.println("number is= "+i);
            i++;
        }
        while(i<=n);
    }
}
```

Output



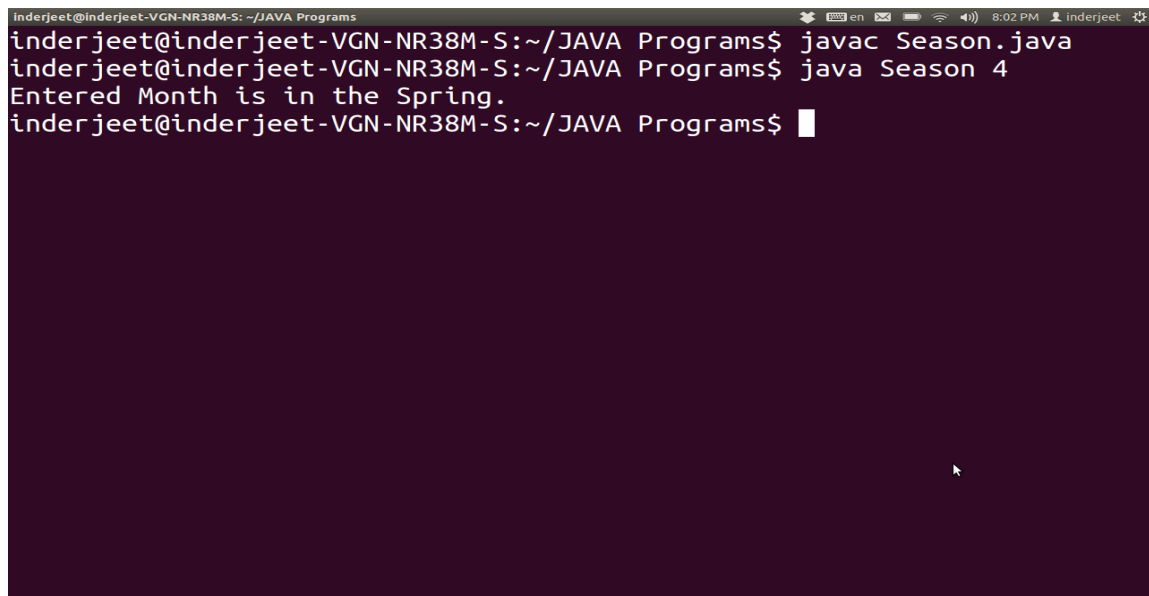
```
inderjeet@inderjeet-VGN-NR38M-S: ~/java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ cd java/
inderjeet@inderjeet-VGN-NR38M-S:~/java$ javac numberDo.java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java numberDo
number is= 1
number is= 2
number is= 3
number is= 4
number is= 5
number is= 6
number is= 7
number is= 8
number is= 9
number is= 10
inderjeet@inderjeet-VGN-NR38M-S:~/java$
```


Practical 6:- Program to determine which season a particular month is in.

Coding

```
class Season {  
    public static void main(String args[]) {  
        int month = Integer.parseInt(args[0]); // April  
        String season;  
        if(month == 12 || month == 1 || month == 2)  
            season = "Winter";  
        else if(month == 3 || month == 4 || month == 5)  
            season = "Spring";  
        else if(month == 6 || month == 7 || month == 8)  
            season = "Summer";  
        else if(month == 9 || month == 10 || month == 11)  
            season = "Autumn";  
        else  
            season = "Bogus Month";  
        System.out.println("April is in the " + season + ".");  
    }  
}
```

Output



```
inderjeet@inderjeet-VGN-NR38M-S: ~/JAVA Programs  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$ javac Season.java  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$ java Season 4  
Entered Month is in the Spring.  
inderjeet@inderjeet-VGN-NR38M-S:~/JAVA Programs$
```

Practical 7:- Program to determine greatest number out of three numbers

Coding

```
import java.util.Scanner;
class largestthree
{
    public static void main(String args[])
    {
        int x, y, z;
        System.out.println("Enter three integers ");
        Scanner in = new Scanner(System.in);
        x = in.nextInt();
        y = in.nextInt();
        z = in.nextInt();
        if ( x > y && x > z )
            System.out.println("First number is largest.");
        else if ( y > x && y > z )
            System.out.println("Second number is largest.");
        else if ( z > x && z > y )
            System.out.println("Third number is largest.");
        else
            System.out.println("Entered numbers are not distinct.");
    }
}
```

Output

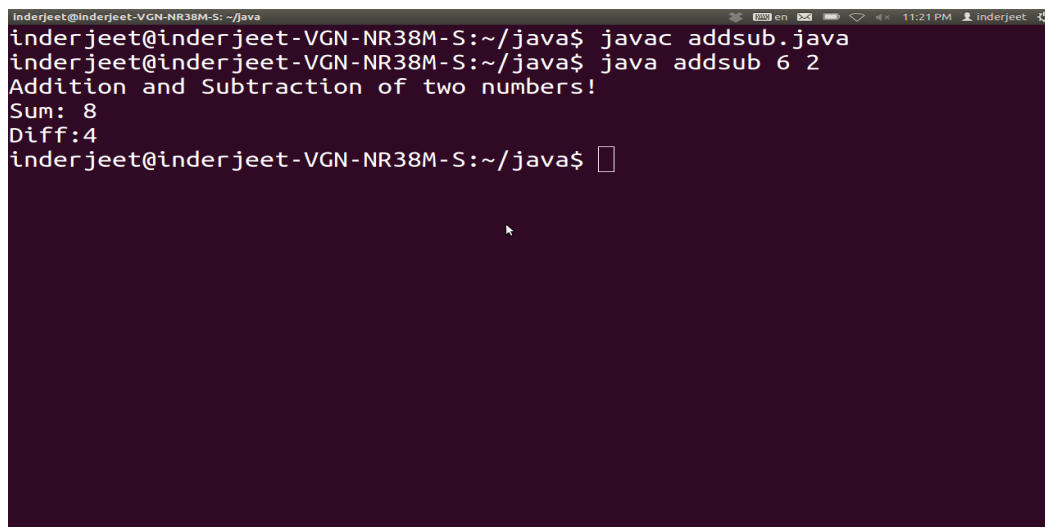
```
inderjeet@inderjeet-VGN-NR38M-S: ~/java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ javac largestthree.java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java largestthree
Enter three integers
28
343
4982
Third number is largest.
inderjeet@inderjeet-VGN-NR38M-S:~/java$
```

Practical 8:- Program to Addition of two numbers and subtraction Algorithm

Coding

```
class addsub
{
    public static void main(String args[]) {
        System.out.println("Addition and Subtraction of two numbers!");
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);
        int sum = a + b;
        int diff = a - b;
        System.out.println("Sum: " + sum);
        System.out.println("Diff: " + diff);
    }
}
```

Output



```
inderjeet@inderjeet-VGN-NR38M-S: ~/java$ javac addsub.java
inderjeet@inderjeet-VGN-NR38M-S: ~/java$ java addsub 6 2
Addition and Subtraction of two numbers!
Sum: 8
Diff: 4
inderjeet@inderjeet-VGN-NR38M-S: ~/java$
```

Practical 9:- Program to find a number is odd or even

Coding

```
class oddeven{
    public static void main(String[] args) {
        int n;
        n = Integer.parseInt(args[0]);
        if (n % 2 == 0) {
            System.out.println("Given number is Even.");
        }
        else {
            System.out.println("Given number is Odd.");
        }
    }
}
```

Output

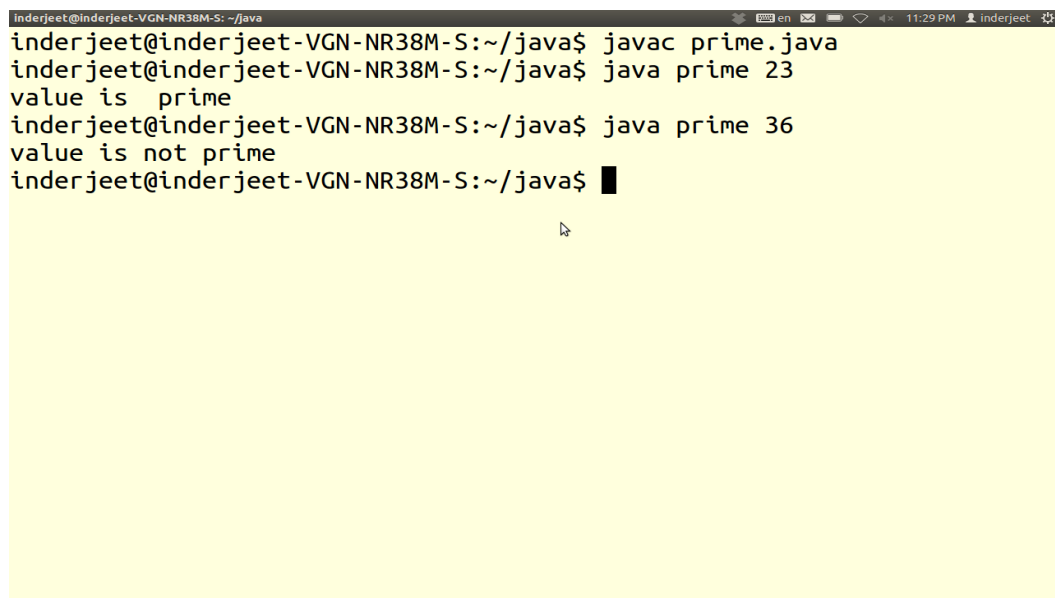
A screenshot of a terminal window with a dark background. The window title is "inderjeet@inderjeet-VGN-NR38M-S: ~/java". The terminal shows the following commands and output:
inderjeet@inderjeet-VGN-NR38M-S:~/java\$ javac oddeven.java
inderjeet@inderjeet-VGN-NR38M-S:~/java\$ java oddeven 3
Given number is Odd.
inderjeet@inderjeet-VGN-NR38M-S:~/java\$
The output "Given number is Odd." is displayed on the third line. The terminal window has a yellow background and a red vertical bar on the right side.

Practical 10:- Program to find a number Prime or not

Coding

```
class prime {  
    public static void main(String args[]) {  
        int num, i;  
        System.out.println("Addition of two numbers!");  
        int num = Integer.parseInt(args[0]);  
        for(i=2; i<num; i++) {  
            int n = num % i;  
            if(n==0) {  
                System.out.println("value is not prime ");  
                break;  
            }  
        }  
        if(i==num) {  
            System.out.println("value is prime ");  
        }  
        if(num==1)  
            System.out.println("value is not prime ");  
    }  
}
```

Output



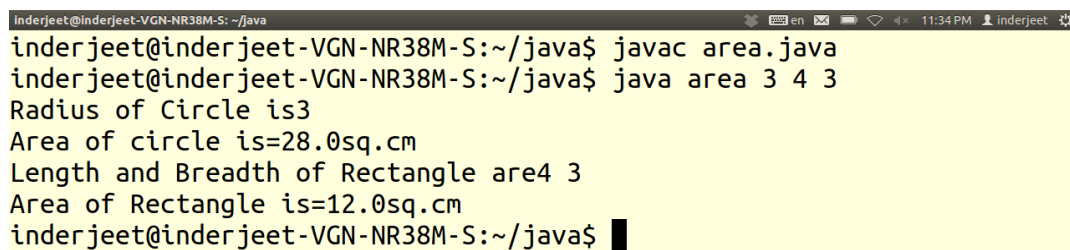
```
inderjeet@inderjeet-VGN-NR38M-S: ~/java  
inderjeet@inderjeet-VGN-NR38M-S:~/java$ javac prime.java  
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java prime 23  
value is prime  
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java prime 36  
value is not prime  
inderjeet@inderjeet-VGN-NR38M-S:~/java$
```

Practical 11:- Program to calculate the area of a circle, rectangle

Coding

```
class area
{
    public static void main(String args[]) {
        int radius=Integer.parseInt(args[0]);
        int len= Integer.parseInt(args[1]);
        int brea= Integer.parseInt(args[0]);
        double area, area_rec;
        area=(22*radius*radius)/7;
        area_rec= len*brea;
        System.out.println("Radius of Circle is" + radius);
        System.out.println("Area of circle is="+area+" "+"sq.cm");
        System.out.println("Length and Breadth of Rectangle are" + len+" "+ brea);
        System.out.println("Area of Rectangle is="+area_rec+" "+"sq.cm");
    }
}
```

Output



```
inderjeet@inderjeet-VGN-NR38M-S: ~/java$ javac area.java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java area 3 4 3
Radius of Circle is3
Area of circle is=28.0sq.cm
Length and Breadth of Rectangle are4 3
Area of Rectangle is=12.0sq.cm
inderjeet@inderjeet-VGN-NR38M-S:~/java$
```

Practical 12:- Program to find average of n numbers using arrays

Coding

```
class avg
{
    public static void main(String args[]) {
        double nums[]={1.0,2.0,3.0,4.5,5.6,8.9};
        double result=0.0;
        int i=0;
        for(i=0; i<nums.length; i++) {
            result=result+nums[i];
        }
        double average=result/nums.length;
        System.out.println("Average value is "+average);
    }
}
```

Output



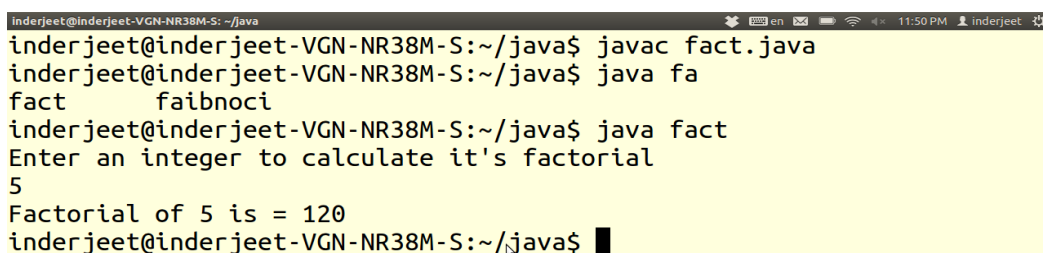
```
inderjeet@inderjeet-VGN-NR38M-S: ~/java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ javac avg.java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java avg
Average value is 4.166666666666667
inderjeet@inderjeet-VGN-NR38M-S:~/java$
```


Practical 13:- Program to find factorial of a number

Coding

```
import java.util.Scanner;
class fact
{
    public static void main(String args[]) {
        int n, c, fact = 1;
        System.out.println("Enter an integer to calculate it's factorial");
        Scanner in = new Scanner(System.in);
        n = in.nextInt();
        if ( n < 0 )
            System.out.println("Number should be non-negative.");
        else {
            for ( c = 1 ; c <= n ; c++ )
                fact = fact*c;
            System.out.println("Factorial of "+n+" is = "+fact);
        }
    }
}
```

Output



```
inderjeet@inderjeet-VGN-NR38M-S: ~/java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ javac fact.java
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java fa
fact      faibnoci
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java fact
Enter an integer to calculate it's factorial
5
Factorial of 5 is = 120
inderjeet@inderjeet-VGN-NR38M-S:~/java$
```

Program 14:- Program to find perimeter of a circle using final keyword

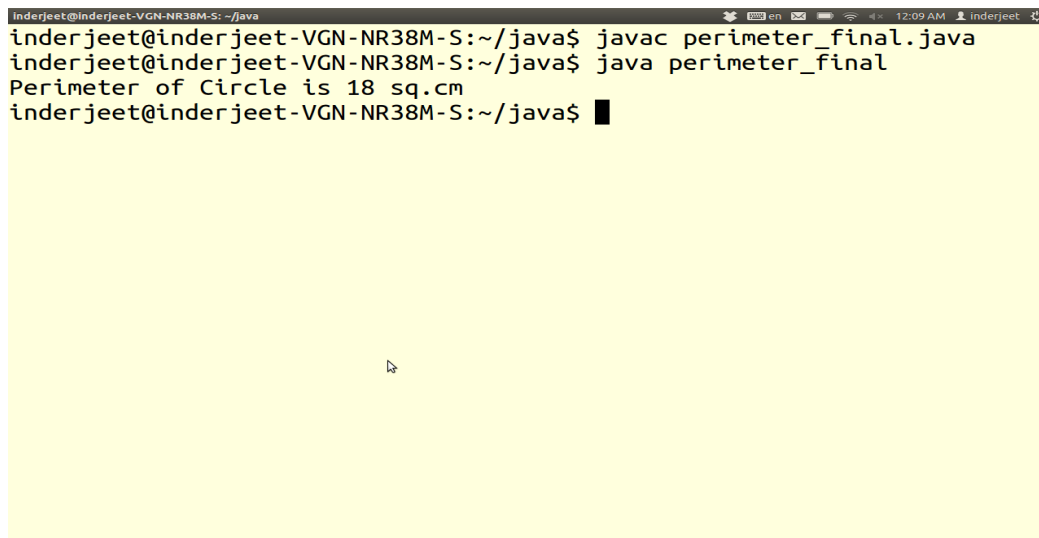
Coding

```
abstract class perimeter{
    int radius;
    abstract void set(int r);
    abstract void peri_meter();
}

class perimeter_circle extends perimeter{
    final void set(int r){
        this.radius=r;
    }
    final void peri_meter(){
        System.out.println("Perimeter of Circle is " + ((2*22*radius)/7)+ " sq.cm"); }
}

class perimeter_final{
    public static void main (String args[]){
        perimeter_circle peri = new perimeter_circle();
        peri.set(3);
        peri.peri_meter(); }
}
```

Output

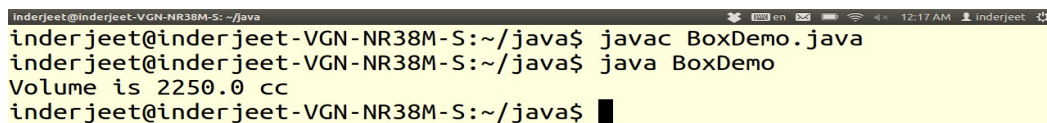


Practical 15:- Program which does the creation of class and object .Algorithm to find volume of a box

Coding

```
class Box {  
    double width;  
    double height;  
    double depth;  
}  
// This class declares an object of type Box.  
class BoxDemo {  
    public static void main(String args[]) {  
        Box mybox = new Box();  
        double vol;  
        // assign values to mybox's instance variables  
        mybox.width = 10;  
        mybox.depth = 15;  
        mybox.height = 15;  
        // compute volume of box  
        vol = mybox.width * mybox.height * mybox.depth;  
        System.out.println("Volume is " + vol);  
    } }  
}
```

Output



```
inderjeet@inderjeet-VGN-NR38M-S: ~/java  
inderjeet@inderjeet-VGN-NR38M-S:~/java$ javac BoxDemo.java  
inderjeet@inderjeet-VGN-NR38M-S:~/java$ java BoxDemo  
Volume is 2250.0 cc  
inderjeet@inderjeet-VGN-NR38M-S:~/java$
```

Practical 16:- Program to find volume of a box using constructor

Coding

```
/* Here, Box uses a constructor to initialize the
dimensions of a box.
*/
class Box {
    double width;
    double height;
    double depth;
// This is the constructor for Box.
    Box() {
        System.out.println("Constructing Box");
        width = 10;
        height = 10;
        depth = 10;
    }
// compute and return volume
    double volume() {
        return width * height * depth;
    }
}
class BoxConst {
    public static void main(String args[]) {
// declare, allocate, and initialize Box objects
        Box mybox1 = new Box();
        Box mybox2 = new Box();
        double vol;
// get volume of first box
        vol = mybox1.volume();
        System.out.println("Volume is " + vol);
// get volume of second box
        vol = mybox2.volume();
        System.out.println("Volume is " + vol);
    }
}
```

} }

Output

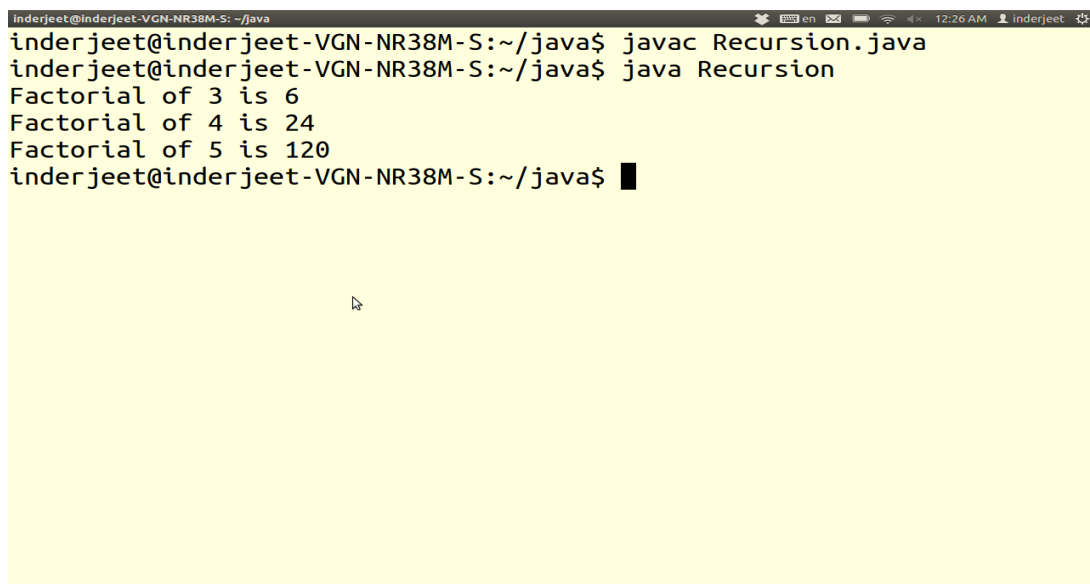
```
inderjeet@inderjeet-VGN-NR38M-S: ~/java$ javac BoxConst.java
inderjeet@inderjeet-VGN-NR38M-S: ~/java$ java BoxConst
Constructing Box
Constructing Box
Volume is 1000.0
Volume is 1000.0
inderjeet@inderjeet-VGN-NR38M-S: ~/java$
```

Practical 17:- To find factorial of a number by passing reference of class factorial to the recursion class.

Coding

```
class Factorial {  
    // this is a recursive function  
    int fact(int n) {  
        int result;  
        if(n==1) return 1;  
        result = fact(n-1) * n;  
        return result;  
    }  
}  
  
class Recursion {  
    public static void main(String args[]) {  
        Factorial f = new Factorial();  
        System.out.println("Factorial of 3 is " + f.fact(3));  
        System.out.println("Factorial of 4 is " + f.fact(4));  
        System.out.println("Factorial of 5 is " + f.fact(5));  
    }  
}
```

Output



```
inderjeet@inderjeet-VGN-NR38M-S: ~/java$ javac Recursion.java  
inderjeet@inderjeet-VGN-NR38M-S: ~/java$ java Recursion  
Factorial of 3 is 6  
Factorial of 4 is 24  
Factorial of 5 is 120  
inderjeet@inderjeet-VGN-NR38M-S: ~/java$ █
```