

**BLOOMINGDALE INTERNATIONAL SCHOOL (ICSE)**

**IRRIGATION COLONY, NARAYANGAON**

**Computer Project**

**I Term Exam**

**2014-15**

**Project By-**

**Atharv B. Darekar**

**X**

**AO-011**

**Guided By-**

**Mr. S. Hakke**

**Short history about JAVA language.**

Java is a [computer programming language](http://en.wikipedia.org/wiki/Computer_programming_language) that is [concurrent](http://en.wikipedia.org/wiki/Concurrent_computing), [class-based](http://en.wikipedia.org/wiki/Class-based), [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible. Java applications are typically [compiled](http://en.wikipedia.org/wiki/Compiler) to [byte code](http://en.wikipedia.org/wiki/Java_bytecode) ([class file](http://en.wikipedia.org/wiki/Class_(file_format))) that can run on any [Java virtual machine](http://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computer architecture](http://en.wikipedia.org/wiki/Computer_architecture). Java is, as of 2014, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by [James Gosling](http://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) (which has since [merged into Oracle Corporation](http://en.wikipedia.org/wiki/Sun_acquisition_by_Oracle)) and released in 1995 as a core component of Sun Microsystems' [Java platform](http://en.wikipedia.org/wiki/Java_(software_platform)). The language derives much of its [syntax](http://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](http://en.wikipedia.org/wiki/C_(programming_language)) and [C++](http://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](http://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. The original and [reference implementation](http://en.wikipedia.org/wiki/Reference_implementation_(computing)) Java [compilers](http://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](http://en.wikipedia.org/wiki/Library_(computing)) were developed by Sun from 1991 and first released in 1995.

[Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) released the first public implementation as Java 1.0 in 1995. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major [web browsers](http://en.wikipedia.org/wiki/Web_browser) soon incorporated the ability to run [Java applets](http://en.wikipedia.org/wiki/Java_applet) within web pages, and Java quickly became popular.

[James Gosling](http://en.wikipedia.org/wiki/James_Gosling), Mike Sheridan, and [Patrick Naughton](http://en.wikipedia.org/wiki/Patrick_Naughton) initiated the Java language project in June 1991.Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.The language was initially called [Oak](http://en.wikipedia.org/wiki/Oak_(programming_language)) after an [oak](http://en.wikipedia.org/wiki/Oak) tree that stood outside Gosling's office, it went by the name Green later, and was later renamed Java, from [Java coffee](http://en.wikipedia.org/wiki/Java_coffee), said to be consumed in large quantities by the language's creators.Gosling aimed to implement a [virtual machine](http://en.wikipedia.org/wiki/Virtual_machine) and a language that had a familiar [C](http://en.wikipedia.org/wiki/C_(programming_language))/[C++](http://en.wikipedia.org/wiki/C%2B%2B) style of notation. On November 13, 2006, Sun released much of Java as [free and open source software](http://en.wikipedia.org/wiki/Free_and_open_source_software), (FOSS), under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL). On May 8, 2007, Sun finished the process, making all of Java's core code available under [free software](http://en.wikipedia.org/wiki/Free_software)/open-source distribution terms, aside from a small portion of code to which Sun did not hold the copyright. Sun's vice-president Rich Green said that Sun's ideal role with regards to Java was as an "evangelist."  There are 930 million [Java Runtime Environment](http://en.wikipedia.org/wiki/Java_(software_platform)) downloads each year and 3 billion [mobile phones](http://en.wikipedia.org/wiki/Mobile_phone) run Java.On April 2, 2010, James Gosling resigned from Oracle.

**Steps for creating object.**

There are three steps for creating an object, they are as follow:

* Declaration: It uses class as data type along with an object.
* Instantiation : The “new” Keyword is used for allocating an object in dynamic memory.
* Initialization: It is used for calling a constructor to initialize attributes of an object

With reference to the above illustration , the process of creating an object of the class ‘Rainbow’ is written as:

Rainbow violet =new rainbow ();

Rainbow violet = new rainbow();

↑ ↑ ↑ ↑

Class object operator constructor

**Steps for compiling and executing.**

For compilation we have to follow following steps

* After completion of program there is an icon named as compile this will compile the program.

Execution of program:

* Select the icon of the class and click the right button on the mouse.
* A drop down menu will appear on the screen , select and click void main (String[] args)
* A method call window appears on the screen click ok.

**key words.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| case | switch | else | byte | Import | goto |
| Do | const | throws | public | package | private |
| Try | int | boolean | break | class | default |
| For | while | double | float | static | catch |
| Long | if | new | void | char | short |

**Importance of ternary operator.**

Ternary Operator (Conditional assignment) Ternary operator deal with three operands. It is also called as conditional assignment statement because the value assigned to variable depending upon logical expression example:

|  |
| --- |
| int t;  t= (4>6) ? 56 : 43 ;  System.out.println(" the result is ="+t);  [Here since the statement is false thus it will print 43] |

**Mathematical functions:**

1. **Math.sqrt():-**

This function is used to find the square root of a positive number. It returns a double type value.

**Example:-**

double n=Math.sqrt(4,0);

It returns a double type value for n as 2.0.

1. **Math.min():-**

This function returns minimum of two numbers. The return value depends on the values used as the argument of the function.

**Example:-**

int n=Math.min (4, 6);

It returns an integer type value of n as 4 which is minimum of 4 and 6.

1. **Math.max():-**

This function is used to find the maximum of two given arguments. It returns a value depending upon the arguments.

**Example:-**

int n=Math.max (4, 8);

It returns an integer type value of n as 4 which are maximum of 4 and 8.

1. **Math.pow():-**

This function is used to find the power rose to a specified base. It always returns a double type data.

**Example:-**

double d=Math.pow(2.0,3.0);

It returns a double type value for d as 2.0=8.0.

1. **Math.log():-**

This function returns natural logarithmic value of a given argument. It always returns a double type value.

**Example:-**

double x=Math.log(6.25);

It returns a double type value for x as 1.8325.

1. **Math.abs():-**

This function returns absolute value i.e, only magnitude of the number. It returns int/long/double value depending upon the argument supplied.

**Example:-**

double d=Math.abs(6.25);

It returns a double type value for x as 3.66.

int n=Math.abs (-8);

It returns an integer type value for n as 8.

1. **Math.round():-**

This function returns the value of a given argument in a rounded form. It always returns a double type value. If the fractional part of the number is below 0.5, it returns the same value, otherwise it returns the next integer value in double data type.

**Example:-**

double n=Math.round(6.25);

It returns an integer type value for n as 6.25.

1. **Math.floor():-**

This function is used to round number. It always returns a number down to the nearest integer.

It always returns a double type value.

**Example:-**

double n=Math.floor (6.25);

1. **Math.ceil():-**

This function also provides the rounded value to the next higher integer. It always returns a double type value.

**Example:-**

double n=Math.ceil (6.25);

It returns an integer type value for n as 7.0.

**10.Math.sin(), Math.cos (), Math.tan():-**

These are the trigonometrical functions. They are used to find the sin, cos, tan values of a given angle in radian as an argument.

**Example:-**

doubled=Math.sin(x);

double d=Math.cos(x);

double d=Math.tan(x);

Here, the argument x is an angle provided in radian. These functions returns double type values.

Generally, the angles are measured in degrees. Thus, it is necessary to convert them from degree to radian to perform the task.

Conversion from degree to radian:-

180º degree = Radian

In order to convert angle from degree to radian use the following formula:-

Radian = 22/ (7\*180))\* Degree [As

To calculate the value of sin30º:-

Let,

int a = 30;

In radian: x=22/ (7\*180))\*a;

Then double, d= Math.sin(x);

It returns the value of sin 30º as 0.05.

**11.Math.asin(), Math.acos (), Math.atan():-**

These are the trigonometrical functions. Each function provides angle corresponding to the given sin, cos and tan.

Degree = (Radian\*7\*180)/22.

**12.Math.exp():-**

This function also provides exponential value. It returns a double type value.

**Example:-**

double n=Math.exp(6.25);

It returns an integer type value for n as 518.0128.

**13.Math.rint():-**

It returns the truncated value of the number when the fractional part is less or equal 0.5, it returns the same value, otherwise it returns the next integer value for more than 0.5 in double data type.

**Example:-**

double d=Math.rint(6.45);

It returns an integer type value for d as 6.0.

**14.Math.random():-**

This function returns a random number between 0 and 1. It returns a double type value. Normally, the return data type is a fractional number.

**Example:-**

double d=Math.random();

It will return any double value to d between 0 and 1.

**Input Methods in Java Language**

As you know that the program is a set of instructions which need some data value to get the result. Java language provides various ways to get the data value within a program. They are as follow:

* By assigning the value
* By using BlueJ system
* By using Input Stream
* By using scanner class
* By using Command line argument

**By assigning the value**

In this segment of Java programming, the students are made to learn by taking values of data of their own or mentioned otherwise within the program along with the appropriate data types.

e.g.

int a=10,b=20,s,d,p;

s=a+b;

d=b-a;

p=a\*b;

System.out.println(“The Sum of 10 and 20 is ”+s);

System.out.println(“The Difference of 20 and 10 is ”+d);

System.out.println(“The Product of 10 and 20 is ”+p);

Output :

The Sum of 10 and 20 is 30

The Difference of 20 and 10 is 10

The Product of 10 and 20 is 200

**By BlueJ System**

This is one of the methods to accept the value from the user at the time of execution of the program. The data types and the variable are declared within main().

i.e. public static void main(<data type><variable name>,<data type><variable name>)

e.g. public static void main(int a, int b)

**By Stream Reader**

Package in Java is basically a collection of classes. Each package includes related built-in functions, which may be used while developing programming logic. In order to develop a program using streams, some of the functions related with input/output operations are invoked.

**Buffer**

CPU or processor is the fastest device in a computer. Other peripheral devices are comparatively slower than processor. Due to speed difference it becomes difficult to have data communication between processor and peripheral devices. Hence, a high-speed memory is applied between I/O devices and processor used as a bridge to synchronize their speeds. This high-speed temporary storage (cache memory) is termed as Buffer. You need to activate buffer before any input/output operation.

**Activation of Buffer in Java**

* Declaring a Java package at the beginning of the program. It allows performing all types of input and output tasks during the execution of the program.

**i.e. import java.io.\*;**

* Declaring buffer to store the data values at the time of execution along with the IOException with the main().

**i.e. public static void main(String args[])throws IOException**

**BufferedReader A=new BufferedReader(new InputStreamReader(System.in));**

The function is so special that it is implicitly invoked as the command is issued to execute a program. The statement **‘throws IOException’** eliminates I/O errors in the program. It passes the report on I/O errors to the exception handler of Java System.

* Declaring a relevant message in order to enable input editor:

i.e. System.out.pritnln(“ //MESSAGE// ”);

**Syntax to accept data of various type:**

1. **Numerical Values:**

To accept int, long, float and double value

<variable>=Integer.parseInt(A.readLine());

<variable>=Long.parseLong(A.readLine());

<variable>=Float.parseFloat(A.readLine());

<variable>=Double.parseDouble(A.readLine());

1. **Character and String**

To accept char and String

<variable>=(char) (A.readLine());

<variable>=A.readLine();

**By Scanner Class**

Scanner class is available in system package java.util. You have to import java.util package to perform the different tasks in scanner class. The process to import java.util package is as shown : **import java.util.\*; or import java.util.scanner;**

**Syntax to accept data of various type**

**Scanner A=new Scanner(System.in);**

1. **Numerical Values:**

To accept int, long, float and double value

<variable>=A.nextInt();

<variable>=A.nextLong();

<variable>=A.nextFloat ();

<variable>=A.nextDouble();

1. **Character and String**

To accept char and String

<variable>=A.next();

<variable>=A.nextLine();

**By Command Line Argument**

This is one of the way to accept the data values from the user and sends the argument to the main . It doesn’t require to import package or activate buffer. While accepting the value from console, the system stores the data values in different locations as arrays of strings. The size of the array can be determined by using the built in function args.length. The argument to the main() are passed through args[0], args[1],args[2]….

**Syntax to accept data of various type**

1. **Numerical Values:**

To accept int, long, float and double value

<variable>=Integer.parseInt(args[0]);

<variable>=Long.parseLong(args[0]);

<variable>=Float.parseFloat(args[0]);

<variable>=Double.parseDouble(args[0]);

1. **Character and String**

To accept char and String

<variable>= (args[0]). (char) at(0);

<variable>= args[0];

**SCOPE OF VARIABLE**

You can use a variable to perform various operations. A variable declared in a block exists only within the visibility of that block. In the program snippet shown above the variable x is declared under if block. Hence, it remains active till the end of the internal block. It is invalid to apply x outside the internal block. Variables a, b and c exist throughout the program unless final closing bracket is applied.

**For Example:** Class scope {

Public static void main (String args []) {

int a=4; b=8; c;

//Variables can be used up to closing bracket of this block

c= a+b;

if (c>10)

{

int x;

//Variable x can be used only within this block

x= 2\*c;

}

System.out.println(x);

//Variable x is not applied here i.e, invalid

System.out.println(c);

//Variable c is applied here

}

}

**NESTED IF-ELSE**

When an if statement is placed within another if, it is known as nested if statement. It can be done by using if-else-if structure.

**Syntax:**

if (condition 1)

{

if (condition 2)

Statement 1

else

Statement 2

}

else

{

if (condition 3)

Statement 3

else

Statement 4

}

**Explanation:**

In the syntax shown above, first of all condition 1 is checked, if it is true then it further checks condition 2 and performs action statement 1 or statement 2 accordingly.

In case condition 1 is false, the control enters else part and further checks condition 3. Statement 3 and statement 4 are executed according to the result of condition 3.

**e.g. Structure of nested-if-else**

if(a>b) //Condition 1

{

If(a>c) //Condition 2

max=a; //Statement 1

else

max=c; //Statement 2

}

else

{

if(b>c) //Condition 3

{

max=b; //Statement 3

else

max=c; // Statement 4

}

System.out.println(“Maximum =”+max);

Statement 2

False

Check Condition 2

Statement 1

True

True

Statement 3

Statement 4

False

Check Condition 3

False

True

Check Condition 1

**P1.WAP To find difference between Simple Interest and Compound Interest when principle, rate, time.**

import java.io.\*;

public class Interest

{

public static void main (String args[]) throws IOException

{

BufferedReader A=new BufferedReader (new InputStreamReader(System.in));

double p,r,t,SI,AM,CI,AM2,P;

System.out.println("Enter an value for Principle (Rs.) 'p'=");

p=Double.parseDouble(A.readLine());

System.out.println("Enter an value for rate p.a % 'r'=");

r=Double.parseDouble(A.readLine());

System.out.println("Enter an value for time span (years) 't'=");

t=Double.parseDouble(A.readLine());

SI=p\*r\*t/100;

System.out.println("Simple Interest =\n Rs.="+SI);

AM=SI+p;

System.out.println("Amount on Simple Interest=\n Rs."+AM);

AM2=p\*(Math.pow((1+r/100),t));

System.out.println("Amount for Compound Interest =\n Rs."+AM2);

CI=AM2-p;

System.out.println("Compound Interest =\n Rs. "+SI);

P=CI-SI;

System.out.println("Profit on using Compound Intrest=\n Rs."+P);

}

}

**Output:-**

Enter an value for Principle (Rs.) 'p'=

10000

Enter an value for rate p.a % 'r'=

2.5

Enter an value for time span (years) 't'=

2

Simple Interest =

Rs.=500.0

Amount on Simple Interest=

Rs.10500.0

Amount for Compound Interest =

Rs.10506.25

Compound Interest =

Rs. 500.0

Profit on using Compound Intrest=

Rs.6.25

**P2.WAP to Calculate Area, Perimeter and diagonal of rectangle.**

import java.io.\*;

import java.math.\*;

public class Rectangle

{

public static void main (String args[])throws IOException

{

BufferedReader A=new BufferedReader (new InputStreamReader(System.in));

int l,b,a,p;

double d;

System.out.println("Enter Value for Length");

l=Integer.parseInt(A.readLine());

System.out.println("Enter Value for Bredth");

b=Integer.parseInt(A.readLine());

a=l\*b;

System.out.println("Area of the given Rectangle =\t"+a);

p=2\*(l+b);

System.out.println("Perimeter of Rectangle =\t"+p);

d=Math.sqrt(l\*l+b\*b);

System.out.println("Diegonal of the given Rectangle =\t"+d);

}

}

**Output:-**

Enter Value for Length

50

Enter Value for Bredth

20

Area of the given Rectangle = 1000

Perimeter of Rectangle = 140

Diegonal of the given Rectangle = 53.85164807134504

**P3.WAP to accept number of days and display year, month and days.**

class Converting\_Days\_to\_Year\_Month\_n\_Days

{

public static void main(int Number\_Of\_Days)

{

int buffer,year,month,days;

year=Number\_Of\_Days/365;

buffer=Number\_Of\_Days%365;

month=buffer/30;

days=buffer%30;

System.out.println("The number of Years is :-\t\t"+year);

System.out.println("The number of months is :-\t\t"+month);

System.out.println("The number of Days is :-\t\t"+days);

}

}

**Output:-**

Input= 560

The number of Years is :- 1

The number of months is :- 6

The number of Days is :- 15

**P4.WAP to display Armstrong number 1 to 1000.**

class Armstrong\_1\_to\_1000

{

public static void main (String [] args)

{

for (int k=1;k<=1000;k++)

{

int n=k,d=0,s=0;

while(n>0)

{

d = n % 10;

s = s+(d \* d \* d);

n = n/10;

}

if (s==k)

{

System.out.println (k+" is Armstrong number");

}

}

}

}

**Output:-**

1 is Armstrong number

153 is Armstrong number

370 is Armstrong number

371 is Armstrong number

407 is Armstrong number

**P5.WAP to display Pattern**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

public class STAR\_Pyramid

{

public static void main(String args[])

{

for(int i=1; i<= 5 ;i++)

{

for(int j=0; j < i; j++)

{

System.out.print("\*");

}

System.out.println("");

}

}

}

**P6.WAP to check weather enter number is Palindrome or not.**

import java.io.\*;

public class Palindrome

{

public static void main(String args[ ]) throws IOException

{

BufferedReader A= new BufferedReader(new InputStreamReader(System.in));

int d,n,r,p;r=0;

System.out.println("Enter a number");

n=Integer.parseInt(A.readLine());

p=n;

do

{

d=n%10;

r=r\*10+d;

n=n/10;

}

while(n!=0);

if(r==p)

System.out.println("Palindrome Number");

else

System.out.println("Not a Pslindrome Number");

}

}

**Output:-**

Enter a number

121

Palindrome Number

**P7.WAP to display Pattern**

**1**

**12**

**123**

**1234**

**12345**

**123456**

class PATTERN\_Piramid\_Numbers

{

public static void main(String[] args)

{

for(int i=1; i<= 6 ;i++)

{

for(int j=0; j < i; j++)

{

System.out.print(j+1);

}

System.out.println("");

}

}

}

**P8.To identify weather the entered number is Prime or not.**

import java.util.\*;

public class Prime\_no\_Identifier

{

public static void main(String args[])

{

Scanner A=new Scanner(System.in);

doublein,a,b=0;

System.out.println("Enter a Number");

in=A.nextInt();

for(a=1;a<=in;a++)

{

if(in%a==0)

b=b+1;

}

if(b==2)

System.out.println("Its a Prime Number");

else

System.out.println("Its Not a Prime Number");

}

}

**Output:-**

Enter a Number

5

Its a Prime Number

**P9.WAP to display Pattern**

**\* \* \* \***

**\* \***

**\* \***

**\* \* \* \***

public class PATTERN\_Border

{

public static void main (String args[])

{

inti,j;

for (i=1;i<=4;i++)

{

for(j=1;j<=4;j++)

{

if (i==1||j==1||i==4||j==4)

System.out.print("\*");

else

System.out.print(" ");

}

System.out.println(" ");

}

}

}

**P10.WAP to display day by using switch case**

import java.util.\*;

public class Switch\_Case

{

public static void main(String args[])

{

Scanner A=new Scanner(System.in);

System.out.println("Enter any Number From 1 to 7");

String in=(A.nextLine());

switch(in)

{

case "1":

System.out.println("Today is Monday");

break;

case "2":

System.out.println("Today is Tuesday");

break;

case "3":

System.out.println("Today is Wednesday");

break;

case "4":

System.out.println("Today is Thursday");

break;

case "5":

System.out.println("Today is Friday");

break;

case "6":

System.out.println("Today is Saturday");

break;

case "7":

System.out.println("Today is Sunday");

break;

default:

System.out.println("Invalid Option");

}

}

}

}

**Output:-**

Enter any Number From 1 to 7

5

Today is Friday

**P11.WAP to calculate if Laptop has 35 % and Printer has 25 % discount and MRP is taken as input for Laptop and Printer.**

public class Offer

{

public static void main (intc,int p)

{

int r1=15,r2=10;

double d1,d2,m=0,n=0;

d1=(double)r1/100\*c;

d2=(double)r2/100\*p;

m=c-d1;

n=p-d2;

System.out.println("The price of Laptop after discount=\t"+m);

System.out.println("The price of Printer after discount=\t"+n);

}

}

**Output:-**

Input is MRP of Laptop is 500000 and Printer 12500

The price of Laptop after discount= 42500.0

The price of Printer after discount= 11250.0

**P12.WAP to display Pattern**

**12345**

**22345**

**33345**

**44445**

**55555**

public class PATTERN\_12345\_22345\_33345\_44445\_55555

{

public static void main(String args[])

{

inta,b,c,p=2;

for(a=1;a<=5;a++)

{

for(b=1;b<=a;b++)

System.out.print(a);

for(c=p;c<=5;c++)

System.out.print(c);

System.out.println();

p=p+1;

}

}

}

**P13.WAP to display Pattern**

**1234567**

**12345**

**123**

**1**

public class PATTERN\_1234567\_12345\_123\_1

{

public static void main(String args[])

{

inti,j;

for(i=7;i>=1;i-=2)

{

for(j=1;j<=i;j++)

System.out.print(j+"");

System.out.println();

}

}

}

**P14.WAP to display Pattern**

**666666**

**55555**

**4444**

**333**

**22**

**1**

public class PATTERN\_666666\_55555\_4444\_333\_22\_1

{

public static void main(String args[])

{

for(int i=6;i>=1;i--)

{

for(int j=1;j<=i;j++)

{

System.out.print(i);

}

System.out.println();

}

}

}

**P15.WAP to display Average, Total and Percentage of a Student**

public class D

{

public static void main(String args[])

{

System.out.println("Student Name- Atharv B. Darekar \n Roll no.- AO-011 \n Std- X");

float math=96,comp=192,phy=94,chem=96,bio=94;

System.out.println("");

System.out.println("Marks Obtained in");

System.out.println("Mathematics "+math+" /100.0");

System.out.println("Physics "+phy+" /100.0");

System.out.println("Chemistry "+chem+" /100.0");

System.out.println("Biology "+bio+" /100.0");

System.out.println("Computer "+comp+" /200.0");

System.out.println("");

System.out.println("Total Marks ="+(math+comp+phy+chem+bio)+" /600.0");

System.out.println("");

System.out.println("Average Marks ="+((math+phy+chem+bio)/5));

System.out.println("");

System.out.println("Percentage Obtained Out of 420.0 ="+(((math+comp+phy+chem+bio)/600)\*100));

}

}

**Output:-**

Student Name- Atharv B. Darekar

Roll no.- AO-011

Std- X

Marks Obtained in

Mathematics 96.0 /100.0

Physics 94.0 /100.0

Chemistry 96.0 /100.0

Biology 94.0 /100.0

Computer 192.0 /200.0

Total Marks =572.0 /600.0

Average Marks =76.0

Percentage Obtained Out of 420.0 =95.33333

**P16.WAP to display pattern**

**1**

**10**

**101**

**1010**

**10101**

public class PATTERN\_1\_10\_101\_1010\_10101

{

public static void main(String args[])

{

int in=1,o1=0,o2=1;

for(int i=1;i<=5;i++)

{

if(i==3||i==5)

o1=in\*o2+1;

else

o1=in\*o2;

o2=10;

in=o1;

System.out.println(o1);

}

}

}