**VARIABLE**:Variable is an name and memory location in order to retrieve the data is called variable.variable is an also known as identifier.

**Ex:int x =10;**

**Here int =datatype**

**X= variable name**

**10=variable name**

**PROGRAM:**

package mani;

public class variable {

public static void main(String[] args) {

int a=10;

System.***out***.println("the integer value="+a);

}

}

**OUTPUT:**

the integer value=10

**OPERATOR:Operator is an specify what operation is an perform an operation.**

**Operation is an symbol of perform an operation**

**OPERAND:**

**Operand is an data which is an particular perform of operation.**

**Operand is an based on three types**

**They are:**

**1.binary operator**

**2.unary operator**

**3.ternary operator**

**1.Binary operator:**

**Binaray operator is an two operand perform particular task is called binary operator**

**Binary operators**

**1.Arithematic operators:**

**1.(+) :addition is combine the two value is called addition**

**2.(-):substraction is an minus of the two values is called substraction**

**3.(\*):multiplication is an used for the multiply into two values**

**4.(/):division is used for divided by an two values.**

**5.(%):moulo division is an used for the remainder is an taken only**

**Program of arithematic opearor:**

package mani;

public class arithematicoperator {

public static void main(String[] args) {

int a=10;

int b=20;

System.***out***.println("the addition value"+(a+b));

System.***out***.println("the substraction value"+(a-b));

System.***out***.println("the multiplication value"+(a\*b));

System.***out***.println("the Division value"+(a/b));

System.***out***.println("the modulo Division value"+(a%b));

}

}

**Output:**

the addition value30

the substraction value-10

the multiplication value200

the Division value0

the modulo Division value10

**RELATIONAL OPRARATORS:Relational operators is an given the output eighther true or false**

**Relational operator are**

**1.>:greterthan**

**2.<:lessthan**

**3.>=:greaterthan equal to**

**4.<=:lessthan equal to**

**4.==:double equal to**

**5.!=:not equal to**

**Program for realational operators:**

package mani;

public class relationaloperators {

public static void main(String[] args) {

int a=10;

int b=20;

System.***out***.println("the greaterthan="+(a>b));

System.***out***.println("the lessthan="+(a<b));

System.***out***.println("the greaterthan equal to="+(a>=b));

System.***out***.println("the less than equal to="+(a<=b));

System.***out***.println("the doble equal to="+(a==b));

System.***out***.println("the greaterthan="+(a!=b));

}

}

**Output:**

the greaterthan=false

the lessthan=true

the greaterthan equal to=false

the less than equal to=true

the doble equal to=false

the greaterthan=true

**3.LOGICAL OPERATORS:**

**There an two types**

**1.and (&&):both true is an true**

**2.or(//):any one true will be true**

**Logical operators are binary operators is an results are coming Boolean**

**NOTE:LOGICAL operator is an only taken for the Boolean data types**

**Program for logical operators:**

package mani;

public class logicaloperators1 {

public static void main(String[] args) {

boolean a=true;

boolean b=false;

System.***out***.println("the logical and="+(a&&b));

System.***out***.println("the logical and="+(a&&a));

System.***out***.println("the logical or="+(a||b));

System.***out***.println("the logical or="+(a||a));

}

}

**Output:**

the logical and=false

the logical and=true

the logical or=true

the logical or=true

**UNARY OPERATOR:  
the unary oerator are two tyepes**

**1.increament operator**

**2.decreament operaor**

**1.increament operator:a++**

**The increamanet operator is an increamnet by 1 is called increamant operaor**

**Two type of increamant operator**

**1.pre increamant operator :++a**

**2.post increament:a++**

**Example program for increament operator:**

public class increamentoperaor {

public static void main(String[] args) {

int a=10;

int b=(a++)+(++a)+(a++);

System.***out***.println("the value of b:"+b);

System.***out***.println(a"the value of a:"+a);

}

}

**Output:**

the value of b:34

the value of a:13

**2.Decreamanet operaor:**

**The decreamnet operator is an decreament by 1 is called dicreament operator**

**The decreamanet operator is an two types**

**1.pre decreament:++a**

**2.post decreament:a++**

**Example program for decreament operator:**

public class increamentoperaor {

public static void main(String[] args) {

int a=10;

int b=(a--)+(--a)+(a--);

System.***out***.println("the value of b:"+b);

System.***out***.println("the value of a:"+a);

}

}

**Output:**

the value of b:26

the value of a:7

**DATA TYPE:**

**Data type is an mechanism of specify what data finding stored the data is called is Datatype**

**Data type is an two types:**

**1.premitive data type**

**2.non primitive datatype**

**1.premitive datatypes types:**

**1.byte**

**2.short**

**3.int**

**4.long**

**4.float**

**5.boolean**

**5.char**

**1.Byte:**

**1Byte=8bits**

**Range stored an Byte**

**Min=-128**

**Max=-127**

**Example [program for byte:**

public class byteoperator {

public static void main(String[] args) {

byte a=10;

byte a=127;

byte a=-128;

/ error:\*byte a=128; here incomatable data type:possible loss from int to byte\*/

//bye a=-130/\* // error:\*byte a=128; here incomatable data type:possible loss from int to byte\*/

//byte a=130;

//byte a=true;//Boolean is an not converted byte

System.***out***.println("the value a="+a);

}

}

**OUTPUT:**

**a=10**

**a=127**

**a=-128**

**Short:**

**2Byte=16bits**

**Range stored an Byte**

**Min=-32768**

**Max=32767**

**Example program:**

public class shortoperator {

public static void main(String[] args) {

short a=10;

short a=-32768;

short a=32767;

/ error:\*byte a=128; here incomatable data type:possible loss from int to byte\*/

//bye a=-32769/\* // error:\*byte a=128; here incomatable data type:possible loss from int to byte\*/

//short a=32769;

//short a=true;//Boolean is an not converted byte

System.***out***.println("The value a="+a);

}

}

**OUTPUT:**

**The value a=10**

**The value a=-32768**

**The value a=32767**

**3.int**

**4Byte=32bits**

**Range stored an Byte**

**Min=-2147483648**

**Max=2147483647**

class shortoperator {

public static void main(String[] args) {

int a=10;

int a=-2147483648;

int a=2147483647;

/ error:\*int a=2147483649; here incompatable data type:possible loss from int to byte\*/

//int a=-2147483649/\* // error:\*byte a=128; here incompatable data type:possible loss from int to byte\*/

//short a=32769;

//short a=true;//Boolean is an not converted byte

System.***out***.println("The value a="+a);

}

}

**OUTPUT:**

(The value a=10

(The value a=-2147483648

(The value a=2147483647+

**3.LONG**

**16Byte=64bits**

**Range stored an Byte**

**Min=**

**Max=**

**Example program:**

**class longdatatype{**

**public static void main(String[]args)**

**{**

**long l=900000000000000l;**

**System.out.println(“the long datatype=”+l);**

**}**

**}**

**Output:**

**The long datatype=900000000000000**

**Float:**

**The float is an decimal number**

**The float is size 4**

**The float is an singleprecisions 4 to 6**

**the float is an primitive datatype**

**the float is an denoted by suffix**

**ex=a=10.34f**

**example program for float:**

**class floatdatatype{**

**public static void main(String[]args)**

**{**

**Float f=10.24f;**

**System.out.println(f);**

**OUTPUT:**

**10.24**

**CHARACTER:**

**Min=0**

**Max=35565**

**THE CHARACTER IS AN taken only one one alphabet with single quoatation.**

**Ex:**

**Class characterdemo{**

**Public static void main(String[]args)**

**{**

**char c=’a’;**

**char a=65;**

**System.out.println(“the display on char value=”+c);**

**System.out.println(“the display of the value=”+a);**

**}**

**}**

**OUTPUT:**

**The display on char value=a**

**The display of the value=A**

**Boolean datatype:**

**The Boolean datatype is an taken eighther true or flase is called Boolean**

**Datatype**

**Class booleandatatype{**

**Public static void main(String[]args)**

**{**

**boolean b=true;**

**boolean d=True; //error**

**Boolean e=False //error**

**boolean c=false;**

**System.out.println(“the Boolean data type=”+b);**

**System.out.println(“the Boolean data type=”+c);**

**}  
}**

**OUTPUT:**

**the Boolean data type=true**

**the Boolean data type=false**

**TYPE CASTING:**

**The typecasting is an changing one datatype to another data type is called typecasting**

**Typecasting are two types:**

**1.implicity**

**2.explicity**

**1.implicity:**

**\*The implicity is an automatically complier execution**

**\*the implicity is an conversion of lower to higher datatype**

**\*the implicity is also known as WIDENING**

**Ex: int = byte**

**2.Explicity:**

**\*The Explicity is an automatically complier execution**

**\*the Explicity is an conversion of higher to lowerdatatype**

**\*the implicity is also known as NARROWING**

**Ex: int = byte**

**EXAMPLE PROGRAM TYPECASTING:**

**class typecasting{**

**public static void main(String[]args)**

**{**

**byte b=20; // WIDENNING**

**int c=b;**

**System.out.println(c);**

**int d=2147483647; //NARROWING**

**byte e=byte(d);**

**System.out.print(c);**

**System.out.println(e);**

**}**

**}**

**OUTPUT:**

20

-1

**USER REQUIREMENT PROGRAMS:**

package mani;

import java.util.Scanner;

public class method1 {

public static void main(String[] args) {

System.***out***.println("Hello, World!");

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

System.***out***.println("enter an a and b value");

int a=scan.nextInt();

int b=scan.nextInt();

int sum=a+b;

System.***out***.println("the value of sum="+sum);

byte c=scan.nextByte();

System.***out***.println("the user of input byte="+c);

System.***out***.println(a-b);

}

}

**Output:**enter an a and b value

45

45

the value of sum=90

4

the user of input byte=4

0

**METHODS:**

**Method is an setoff instsructions/statement perform particular task is called method**

**Methods are 4types**

**1.merthods are doesn’t expect parameters and doesn’t written return value**

**2.methods are doesn’t expect parameters and with return value**

**3.methods are with parameter and without wriiten return any value**

**4.methods are with parameter and with written return value**

**1.NO PARAMETER AND NO RETURN VALUE**

**Example program**

public class method1 {

public static void main(String[] args) {

System.***out***.println("method is an starting");

*greet*();

System.***out***.println("method is an ended");

}

static void greet() {

System.***out***.println("greet method are staring");

System.***out***.println("good afterNoon");

System.***out***.println("greet() is ended");

}

}

**Output:**

method is an starting

greet method are staring

good afterNoon

greet() is ended

method is an ended

**2.WITHOUT PARAMETER AND WITH RETURN VALUE**

public class method1 {

public static void main(String[] args) {

System.***out***.println("method is an starting");

int result= *greet*();

System.***out***.println("the result are="+result);

}

static int greet() {

System.***out***.println("greet method are staring");

int a=10;

int b=20;

int sum=a+b;

return sum;//sum taken integer datatype but method definition take integer only return any value create assignment operator for the method call

}

}

**OUTPUT:**

method is an starting

greet method are staring

the result are=30

**3.PARAMETER BUT NO RETURN VALUE:**

public class method1 {

public static void main(String[] args) {

System.***out***.println("main method is an started");

int a=10;

int b=30;

*add*(a,b);

System.***out***.println("the with parameter and without return value");

}

static void add(int a,int b)

{

int sum=a+b;

System.***out***.println("the value of sum="+sum);

}

}

**OUTPUT:**

main method is an started

the value of sum=40

the with parameter and without return value

**4.WITH PARAMETER AND WITH RETURN VALUE**

**Example program**

public class method1 {

public static void main(String[] args) {

System.***out***.println("main method is an started");

int a=10;

int b=30;

int result=*add*(a,b);

System.***out***.println("the with parameter and without return value"+result);

}

static int add(int a,int b)

{

int sum=a+b;

return sum;//sum is taken intger but method definition must take integer only

}

}

**OUTPUT:**

main method is an started

the with parameter and without return value40

**OBJECTS:**

**OBJECT IS REAL TIME SLVING PROBLEMS WITH PROGRAM**

**Object is an collection instanceous of a class is called object.**

**Object is an new key keyword created an object in heap memory**

**Object :**

**State behaviour**

**Properties action()**

**Attributes memberfunction()**

**Example:window:**

**State=size,width,height,color**

**Member functions=open,close**

**Example program objects:**

**1.1st create an class is an blueprint of the object.**

**2.object is class create an states(attributes)and behavvior**

**3.next write an main method after create new keyword is create on object on heap memory**

package java1;

public class objects {

int id;//states

String name;

double cgpa;

void study() {

System.***out***.println("the studying timing are=3 hours");

}

//behaviour void attendance() {

System.***out***.println("the attendnace=40");

}

public static void main(String[]args) {

objects o1=new objects();

o1.id=1234;

o1.name="subramanyam";

o1.cgpa=90.09;

o1.study();

o1.attendance();

System.***out***.println(o1.id);

System.***out***.println(o1.name);

System.***out***.println(o1.cgpa);

}

}

**Output:**the studying timing are=3 hours

the attendnace=40

1234

subramanyam

90.09

**Constructor:**

**Constructor is an same class create an obeject initaialize state of the object is called constructor.there is no return type..**

* **Constructor using this keyword that means create an current object;**

**The object and constructor same values but not execution but error is Shadowing by using this keyword;**

**Example program:**

**class HelloWorld {**

**String name;**

**int id;**

**double cgpa;**

**void study(){**

**System.out.println("the enter a value");**

**}**

**HelloWorld(String name,int id,double cgpa){//constructor no return value**

**this.name=name;**

**this.id=id;**

**this.cgpa=cgpa;**

**}**

**public static void main(String[]args){**

**HelloWorld s1=new HelloWorld("mani",200,30.90);//constructor**

**s1.study();**

**System.out.println(s1.name);**

**System.out.println(s1.id);**

**System.out.println(s1.cgpa);**

**}**

**}**

**OUTPUT:  
the enter a value**

**Mani**

**200**

**30.90**

**Control constructor:**

**1.the flow of the execution the program is called control statement**

**3 types:**

**1.selection statement/conditions statements**

**2.looping/iterarive statements**

**3.jumping statements**

**1.conditional statements:**

**The condition is an true execution on only time is called conditional statements.**

**Types of conditional staements:**

**1.if**

**2.if else**

**3.if else ladder**

**4.Switch-case**

**2.Looping statements:**

**The condition is an true execution on many times is called as lopping statements/iterative satements.**

**1.for**

**2.while**

**3.do-while**

**3.jump:**

**The flow of execution with in loops an Switchsatements**

**1.break**

**2.continue**

**3.Return**

**1.conditional statements:**

**Write an javaprogram to welcome students to the techfest those who are score morethan 90percentage**

**Example program:**

import java.util.Scanner;

public class moretahn90percentageusingifcondition {

public static void main(String[] args) {

System.***out***.println("the condition is an if statement");

Scanner s1=new Scanner(System.***in***);

int a=s1.nextInt();

if(a>90) {

System.***out***.println("the welcome to students on techfest");

}

}

}

**OUTPUT:**

the condition is an if statement

91

the welcome to students on techfest

**2.CHECK PERSON ELIGIBLE FOR VOTING**

import java.util.Scanner;

public class moretahn90percentageusingifcondition {

public static void main(String[] args) {

System.***out***.println("the condition is an if statement");

Scanner s1=new Scanner(System.***in***);

int age=s1.nextInt();

if(age>18) {

System.***out***.println("the eligible for the voting");

}

}

}

**OUTPUT:**

the condition is an if statement

19

the eligible for the voting

**\*EvenOrOdd:**

package java1;

import java.util.Scanner;

public class practiceeveryday {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

System.***out***.println("the eneter the value");

int n=s1.nextInt();

if(n%2==0) {

System.***out***.println("n is an even");

}

else {

System.***out***.println("n is an odd");

}

}

}

**OUTPUT:**

the eneter the value

2

n is an even

**LEPAYEARORNOT:**

package java1;

import java.util.Scanner;

public class practiceeveryday {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

System.***out***.println("the eneter the value");

int n=s1.nextInt();

if(n%4==0) {

System.***out***.println("given number leap year ");

}

else {

System.***out***.println("given number not leap year");

}

}

}

the eneter the value

2012

the character is an leap year

**VowelsOrConsonants:if else**

package java1;

import java.util.Scanner;

public class practiceeveryday {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

System.***out***.println("the eneter the value");

char c=s1.next().charAt(0);

if(c=='a'||c=='e'||c=='i'||c=='o'||c=='u') {

System.***out***.println("the character is an vowels");

}

else {

System.***out***.println("the character is an consonants");

}

}

}

**OUTPUT:**

the eneter the value

a

the character is an vowels

Z

The character is an consonant

**Else if using vowels orConsant:**

**The condition are two or more using conditions is true executed an true statement**

package java1;

import java.util.Scanner;

public class practiceeveryday {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

System.***out***.println("the enter the character");

char c=s1.next().charAt(0);

if(c=='a') {

System.***out***.println("the character is an vowels");

}

else if(c=='e') {

System.***out***.println("the character is an vowels");

}

else if(c=='i') {

System.***out***.println("the character is an vowels");

}

else if(c=='o') {

System.***out***.println("the character is an vowels");

}

else if(c=='u') {

System.***out***.println("the character is an vowels");

}

else {

System.***out***.println("the character is an consonant");

}

}}

**OUTPUT:**

the enter the character

a

the character is an vowels

Z

The character is an consonant

**THE LARGEST VALUE USING ELSE IF STATEMENT**

package java1;

import java.util.Scanner;

public class practiceeveryday {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

int a=s1.nextInt();

int b=s1.nextInt();

int c=s1.nextInt();

if(a>b&&a>c) {

System.***out***.println("The a is an large");

}

else if(b>c) {

System.***out***.println("the b is an large");

}

else {

System.***out***.println("the c is an larges");

}

}

}

**OUTPUT:**

10

20

30

the c is an larges

**SWITCH STATEMENT:**

**1.VOWELS OR CONSONANT:**

package java1;

import java.util.Scanner;

public class practiceeveryday {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

char c=s1.next().charAt(0);

switch(c) {

case 'a':System.***out***.println("the character is an vowels");break;

case 'e':System.***out***.println("the character is an vowels");break;

case 'i':System.***out***.println("the character is an vowels");break;

case 'o':System.***out***.println("the character is an vowels");break;

case 'u':System.***out***.println("the character is an vowels");break;

default:System.***out***.println("the character is an Consonant");

}

}

}

a

the character is an vowels

**2.Rainbowcolor:vibgor**

package java1;

import java.util.Scanner;

public class practiceeveryday {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

char c=s1.next().charAt(0);

switch(c) {

case 'v':System.***out***.println("the violet is an rainbow");break;

case 'i':System.***out***.println("the indicate is an rainbow");break;

case 'b':System.***out***.println("the blue is an rainbow");break;

case 'g':System.***out***.println("the green is an rainbows");break;

case 'o':System.***out***.println("the orange is an rainbow");break;

default:System.***out***.println("the not color in rainbow");

}

}

}

**OUTPUT:**

o

the orange is an rainbow

**2.Looopin/iteration:the condition is true is an executed anmanytimes/the condition is true is an again and again and execution until flase .**

**1.for loop:**

**For loop is an as long as execution of the program condition true executed on more times and condition is false stop the program**

**Syntax:**

**1. initialization**

**2. condition**

**3. inc/dec**

**For(initialization;condition;inc/dec)**

**{**

**Body/Statements**

**}**

**Example program:**

**Write aprogram exection on program “how are you”100times**

public class UsinForPrint1000 {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

for(int i=1;i<=10;i++)

System.***out***.println("how are you");

}

}

**Output:**

how are you

how are you

how are you

how are you

how are you

how are you

how are you

how are you

how are you

how are you

**2.while()**

**While is an condition as long as exection .**

**The condition is an true repeated an executed otherwise condition is an false skipped the body**

**Syntax:**

**1.intialization**

**2.condition**

**3.inc/dec**

**Initialization;ex=int i=2;**

**While(condition){**

**Body/statement**

**Inc/dec**

**}**

**Write an program for using while condition display string an” hello world “**

**to print**

public class WhileUsing {

public static void main(String[] args) {

int i=1;

while(i<=10) {

System.***out***.println("Hello world");

i++;

}

}

}

**OUTPUT:**

Hello world

Hello world

Hello world

Hello world

Hello world

Hello world

Hello world

Hello world

Hello world

Hello world

**3.dowhile()the dowhile is an same as long as execution.**

**it is an condition body after while condtion;**

**Sysntax:**

**intialiazation;**

**do{**

**Body/statement**

**Inc/dec**

**}while(condition);**

**Wite aprogram using to print an Good morning 10 times using dowhile**

public class DoWhile {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int i=1;

do {

System.***out***.println("good morning");

i++;

}while(i<=10);

}

}

**OUTPUT:**

good morning

good morning

good morning

good morning

good morning

good morning

good morning

good morning

good morning

good morning

**for loop:**

**wrap factorial**

**1.first taken intial value 1**

**2.forloop**

**3.fact=fact\*i**

package mani;

import java.util.Scanner;

public class FactoRial {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

System.***out***.println("enter an factorial number");

int n=s1.nextInt();

int fact=1;

for(int i=1;i<=n;i++) {

fact=fact\*i;

System.***out***.println(fact);

}

}

}

**OUTPUT:**

enter an factorial number

5

1

2

6

24

120

**Fibbonacci series:**

**First and second adding is called fibonnaci series**

package mani;

import java.util.Scanner;

public class fibbonacciseries {

public static void main(String[]args) {

Scanner s1=new Scanner(System.***in***);

System.***out***.println("enter an fibbonacci series");

int n=s1.nextInt();

int f=1,s=1,temp;

System.***out***.println(f);

System.***out***.println(s);

for(int i=1;i<=n;i++) {

temp=f;

f=s;

s=s+temp;

System.***out***.println(s);

}

}

}

**Output:**

enter an fibbonacci series

5

1

1

2

3

5

8

13

**Table Calculate;**

**1.the table is an calculate 5\*1=5**

**Here 5=num**

**\*=string**

**1=iterarion**

**(=)=string**

**5=num\*i**

System.***out***.println(n+"\*"+i+"="+(n\*i));

package mani;

import java.util.Scanner;

public class Tablecalculate {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

int n=s1.nextInt();

for(int i=1;i<=10;i++) {

System.***out***.println(n+"\*"+i+"="+(n\*i));

}

}

}

**Output:**

2

2\*1=2

2\*2=4

2\*3=6

2\*4=8

2\*5=10

2\*6=12

2\*7=14

2\*8=16

2\*9=18

2\*10=20

**While Loop;**

**Polindrome:121=121**

**d=n%10 121%10=1 12%10=2 1%10=1**

**sum=sum\*10+d 1\*10+1=11 11\*10+2=112 112\*10+1=1121**

**n=n/10 121/10=12 12/10=1**

package mani;

import java.util.Scanner;

public class polindromeornot {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

int n=s1.nextInt();

int dig,sum=0,temp=n;

while(n>0) {

dig=n%10;

sum=sum\*10+dig;

n=n/10;

System.***out***.println(sum);

}

}

}

**OUTPUT:**

12321

1

12

123

1232

12321

**POLINDROME OR NOT:**

package mani;

import java.util.Scanner;

public class polindromeornot {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

int n=s1.nextInt();

int dig,sum=0,temp=n;

while(n>0) {

dig=n%10;

sum=sum\*10+dig;

n=n/10;

System.***out***.println(sum);

}

if(temp==sum) {

System.***out***.println("Given number is an polindrome");

}

else {

System.***out***.println("Given number is an polindrome");

}

}

}

**Output:**

1

12

123

1232

12321

Given number is an polindrome

**Sum Of the Digits:**

**1234=10**

**dig=n%10 1234%10=4 123%10=3 12%10=2 1%10=1**

**sum=sum+dig 0+4=4 4+3=7 7+2=9 9+1=10**

**n=n/10 1234/10 =123 123/10=12 12/10=1**

**here n%10 is an devide an number taken remainder after sum added an digit after remove the last value continue process condition is false stop the program**

package mani;

import java.util.Scanner;

public class Sumofdigits {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

int n=s1.nextInt();

int i=0,dig,sum=0;

while(n>0) {

dig=n%10;

sum=sum+dig;

n=n/10;

System.***out***.println(sum);

}

}

}

**Output:**

1234

4

7

9

10

**AmstrongNumber:**

**The amstrong number is an divide digit cube is called amstrong**

**Ex:153=1\*1\*1+5\*5\*5+3\*3\*3=153**

**Dig=n%10 153%10=3 15%10=5 1%10=1**

**Sum=sum+d\*d\*d 0+3\*3\*3=27 27+5\*5\*5=152 152+1\*1\*=153**

**N=n/10 153/10=15 15/10=1**

package mani;

import java.util.Scanner;

public class AmstrongNumber {

public static void main(String[] args) {

Scanner s1=new Scanner(System.***in***);

int n=s1.nextInt();

int dig,sum=0;

while(n>0) {

dig=n%10;

sum=sum+dig\*dig\*dig;

n=n/10;

System.***out***.println("Amstrong number"+sum);

}

}

}

153

Amstrong number27

Amstrong number152

Amstrong number153

:

PATTERNS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | \* |  |  |  |  |
| 2 | \* | \* |  |  |  |
| 3 | \* | \* | \* |  |  |
| 4 | \* | \* | \* | \* |  |
| 5 | \* | \* | \* | \* | \* |

The above pattern is an increase an colums print columns denoted by j

i j

1 <= 1

2 <= 1,2,

3  **<=** 1,2,3

4 <= 1,2,3,4

5 <= 1,2,3,4,5

J<=i

Program

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=i) {

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

}

}

System.***out***.println("");

}

}

}

OUTPUT:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 |  |  |  |  |
| 2 | 1 | 2 |  |  |  |
| 3 | 1 | 2 | 3 |  |  |
| 4 | 1 | 2 | 3 | 4 |  |
| 5 | 1 | 2 | 3 | 4 | 5 |

The above pattern is an increase an colums print columns

i j

1 <= 1

2 <= 1,2,

3  **<=** 1,2,3

4 <= 1,2,3,4

5 <= 1,2,3,4,5

J<=i

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=i) {

System.***out***.print(j);

}

}

System.***out***.println("");

}

}

}

**OUTPUT:**

1

12

123

1234

12345

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 |  |  |  |  |
| 2 | 2 | 2 |  |  |  |
| 3 | 3 | 3 | 3 |  |  |
| 4 | 4 | 4 | 4 | 4 |  |
| 5 | 5 | 5 | 5 | 5 | 5 |

The above pattern is an SAME an colums print ROWS

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=i) {

System.***out***.print(i);

}

}

System.***out***.println("");

}

}

}

**OUTPUT:**

1

22

333

4444

55555

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | A |  |  |  |  |
| 2 | A | B |  |  |  |
| 3 | A | B | C |  |  |
| 4 | A | B | C | D |  |
| 5 | A | B | C | D | E |

The above pattern is an increase an colums print columns denoted by j

i j

1 <= A 64+i=65=A

2 <= A,B 64+2=B

3  **<=** A,B,C 64+3=C

4 <= A,B,C,D 64+4=D

5 <= A,B,C,D,E 64

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=i) {

System.***out***.print((char)(64+J));

}

}

System.***out***.println("");

}

}

}

**OUTPUT**

A

AB

ABC

ABCD

ABCDE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | A |  |  |  |  |
| 2 | B | B |  |  |  |
| 3 | C | C | C |  |  |
| 4 | D | D | D | D |  |
| 5 | E | E | E | E | E |

The above pattern is an SAME an colums print ROWS

i j

1 <= A 64+i=65=A

2 <= BB 64+2=B

3  **<=** CCC 64+3=C

4 <= DDDD 64+4=D

5 <= EEEEE 64

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=i) {

System.***out***.print((char)(64+i));

}

}

System.***out***.println("");

}

}

}

**OUTPUT:**

A

BB

CCC

DDDD

EEEEE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | \* | \* | \* | \* | \* |
| 2 | \* | \* | \* | \* |  |
| 3 | \* | \* | \* |  |  |
| 4 | \* | \* |  |  |  |
| 5 | \* |  |  |  |  |

The above pattern is an dicrease an colums print columns denoted by j

I j

1 5 j<=6-i=5

2 4 j<=6=i=4

3 3 j<=6-i=3

4 2 j<=6-i=2

5 1 j<=6-i=1

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=6-i) {

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

}

}

System.***out***.println("");

}

}

}

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 2 | 3 | 4 | 5 |
| 2 | 1 | 2 | 3 | 4 |  |
| 3 | 1 | 2 | 3 |  |  |
| 4 | 1 | 2 |  |  |  |
| 5 | 1 |  |  |  |  |

The above pattern is an dicrease an colums print columns denoted by j

**i<=6-I**

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=6-i) {

System.***out***.print(j);

}

}

System.***out***.println("");

}

}

}

12345

1234

123

12

1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 |  |
| 3 | 3 | 3 | 3 |  |  |
| 4 | 4 | 4 |  |  |  |
| 5 | 5 |  |  |  |  |

The above pattern is an SAME an colums print ROWS

**Condition=j<=6-i**

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=6-i) {

System.***out***.print(i);

}

}

System.***out***.println("");

}

}

}

11111

2222

333

44

5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | A | B | C | D | E |
| 2 | A | B | C | D |  |
| 3 | A | B | C |  |  |
| 4 | A | B |  |  |  |
| 5 | A |  |  |  |  |

**THE ABOVE PATTERN IS a is a srting 64+I colimns are increase but to print colums denoted by j**

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=6-i) {

System.***out***.print((char)(64+j));

}

}

System.***out***.println("");

}

}

}

**OUTPUT**

ABCDE

ABCD

ABC

AB

A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | A | A | A | A | A |
| 2 | B | B | B | B |  |
| 3 | C | C | C |  |  |
| 4 | D | D |  |  |  |
| 5 | E |  |  |  |  |

**The above programs are columns are same but rows are print i**

**Condition;j<=6-i print (char)(64+i)**

public class Practice {

public static void main(String[]args) {

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

for(int j=1;j<=5;j++) {

if(j<=6-i) {

System.***out***.print((char)(64+i));

}

}

System.***out***.println("");

}

}

}

**OUTPUT:**

AAAAA

BBBB

CCC

DD

E

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | E | E | E | E | E |
| 2 | D | D | D | D |  |
| 3 | C | C | C |  |  |
| 4 | B | B |  |  |  |
| 5 | A |  |  |  |  |

**THE ABOVE PATTERN ARE DICREASE ARE COLUMNS BUT USING SAME VALUES PRINT ROWS 1 64+5=E**

public class Practice {

public static void main(String[]args) {

for(int i=5;i>=1;i--) {

for(int j=1;j<=5;j++) {

if(j<=i) {

System.***out***.print((char)(64+i));

}

}

System.***out***.println("");

}

}

}

**OUTPUT:**

EEEEE

DDDD

CCC

BB

A

**HILLS patterns:hills is an shape of prism**

The hills patterns ian taken **middle row after clalculate both left and right side**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 |  |  |  | \* |  |  |  |
| 2 |  |  | \* | \* | \* |  |  |
| 3 |  | \* | \* | \* | \* | \* |  |
| 4 | \* | \* | \* | \* | \* | \* | \* |

**The pattern are**

**I j difference add5 i j difrence all add 3**

**1 <= 4 1 + 4 = 3+i=4**

**2 <= 3 2 + 5 =3+i=5**

**3 <= 2 3+ 6 =3+i=6**

**4 <= 1 4+ 7 =3+i=7**

**CONDITION:j<=5-i&&j<=5-I**

**ARRAYS: arrays is a data structure is stored and fixed sequentialy element group of element is an individual**

**Or**

**1.index**

**2.homogeneous**

**3.fixed size**

**Index is an data structure stored data homogeneous of deals with fixed size is called an arrays**

**Arrays is an declaration**

**Array is an creation**

**Array is an initialization**

**1.Arrays is an declaration:**

**Int[]arr;**

**Here arr=array variable**

**Int=integer**

**[]=array dimensions**

**Array is a creation:**

**Arr=new int[5];**

**Here new is an create an object of the Heap memory**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| **0** | **1** | **2** | **3** | **4** |

**1000 1004 1008 1010 1014**

**Referenece variable is 1000**

**Int 4**

**Index 0**

**4\*0+1000=10000=0 but taken starting is array is zero**

**Index 2**

**4\*2+1000=1008**

**3.Initalization:**

**Int[]arr=new int[5];**

**Arr[0]=1;**

**Arr[1]=2;**

**Arr[2]=3;**

**Arr[3]=4;**

**Arr[4]=5;**

public class array1 {

public static void main(String[] args) {

*array*();

}

static void array() {

int[]arr=new int[4];

arr[0]=1;

arr[1]=2;

arr[2]=3;

arr[3]=4;

System.***out***.println(arr[0]);

System.***out***.println(arr[1]);

System.***out***.println(arr[2]);

System.***out***.println(arr[3]);

}

}

**Output;**

1

2

3

4

**The above program is an is an reduncy program for using for loops;**

**For(int i=0;i<=4;i++)**

**System.out.println(arr[i]);**

1

2

3

4

public class array1 {

public static void main(String[] args) {

*array*();

}

static void array() {

int[]arr=new int[5];

arr[0]=10;

arr[1]=20;

arr[2]=30;

arr[3]=40;

arr[4]=50;

for(int i=0;i<=4;i++) {

System.***out***.println(arr[i]);

}

}

}

10

20

30

40

50

**User requirement of array;**

import java.util.Scanner;

public class OneDimenSional1UserREquirement {

public static void main(String[] args) {

System.***out***.println("enter an number");

int[]arr=new int[5];

*array*(arr);

}

static void array(int[] arr){

Scanner s1=new Scanner(System.***in***);

for(int i=0;i<arr.length;i++) {

arr[i]=s1.nextInt();

System.***out***.println(arr[i]);

}

}

}

**OUTPUT:**

10

20

30

40

50

10

20

30

40

50

**1dimensional using evenprint;**

**Condition:int arr[i]%2==0**

import java.util.Scanner;

public class OneDUsingArray {

public static void main(String[] args) {

//declaration and initialization

int[]arr=new int[5];

Scanner s1=new Scanner(System.***in***);

for(int i=0;i<arr.length;i++) {

arr[i]=s1.nextInt();

System.***out***.println(arr[i]);

}

for(int i=0;i<arr.length;i++) {

if(arr[i]%2==0) {

System.***out***.println("given number is an even");break;

}

}

**Output:**

1

1

3

3

4

4

5

5

6

6

given number is an even

**2.2Dimensional Array**

**The wo dimensional array is an one array and another array is called two dimensional array**

**Syntax:int[][]=new int[][]**

**1st dimension create an one for loop(for(int i=0;i<arr.length;i++)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **10** | **20** | **30** | **40** | **50** | **60** |
| **0** | **1** | **2** | **3** | **4** | **5** |

**2nd dimension create an another for loopfor(int j=0;j<arr[i].length;j++);**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **10** | **20** | **30** | **40** | **50** | **60** |
| **0** | **1** | **2** | **3** | **4** | **5** |

import java.util.Scanner;

public class Arraycreation2Dimensions {

public static void main(String[] args) {

//array declaration and initialization

Scanner s1=new Scanner(System.***in***);

int[][]arr=new int[2][5];

//1st array

System.***out***.println("enter an 1st dimension");

for(int i=0;i<arr.length;i++) {

System.***out***.println("enter an second dimension");

for(int j=0;j<arr[i].length;j++) {

arr[i][j]=s1.nextInt();

}

}

}

}

**OUTPUT:**

enter an 1st Dimension

enter an 2nd dimension

10

20 FOR LOOP USING 1ST ARRAY

30

40

50

enter an 2nd dimension

10

20 FOR LOOP USING 2ND ARRAY

30

40

50

**2DIMENSION USING 1ST ARRAY AND 2ND ARRAY USING ADDITION**

package mani;

import java.util.Scanner;

public class Arraycreation2Dimensions {

public static void main(String[] args) {

//array declaration and initialization

Scanner s1=new Scanner(System.***in***);

int[][]arr=new int[2][5];

//1st array

System.***out***.println("enter an 1st Dimension");

for(int i=0;i<arr.length;i++) {

System.***out***.println("enter an 2nd dimension");

for(int j=0;j<arr[i].length;j++) {

arr[i][j]=s1.nextInt();

}

}

//TRAVERSING

for(int i=0;i<arr.length;i++) {

for(int j=0;j<arr[i].length;j++) {

arr[i][j]=arr[i][j]+arr[i][j];

System.***out***.println(arr[i][j]);

}

}

}

}

**Output:**

enter an 1st Dimension

enter an 2nd dimension

1

2

3

4

5

enter an 2nd dimension

6

1

2

3

45

2

4

6

8

10

12

2

4

6

90

**Jagged array:**

**The jagged array is 1st array is an 3 and 2nd array 2 is called jagged array**

import java.util.Scanner;

public class Jaggedarray {

public static void main(String[] args) {

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

int[][]arr=new int[2][];

arr[0]=new int[3];//jagged array

arr[1]=new int[2];//jagged array

//1st array

System.***out***.println("enter an 1st array");

for(int i=0;i<arr.length;i++) {

//2nd array

System.***out***.println("enter an second array");

for(int j=0;j<arr[i].length;j++) {

arr[i][j]=scan.nextInt();

}

}

}

}

**Output:**

enter an 1st array

enter an second array

1

2

3

enter an second array

5

54

**NONSTATIC:WITHOUT USING STATIC IS CALLED NON STATIC/NONSTATIC USING ONLIY BEHAVIOUR IT SI AN FIXED VALUE**

**STSEPS:**

**1.CREATE AN CLASS**

**1.BEHAVIOUR**

**2.next create an mainmethod**

**3.next create an obeject behaviour class**

**4.calling reference.method name(vraiiable);**

**Example:**

**behaviour**

package practic.kn;

public class NonStaticusingFactorial {

**public int factorial**(int n) {//non static

int fact=1;

for(int i=1;i<=n;i++) {

fact=fact\*i;

}

return fact;

}

}

**Mainmmethod:**

package practic.kn;

import java.util.Scanner;

public class Factor {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Scanner s1=new Scanner(System.***in***);

int n= s1.nextInt();

//object

NonStaticusingFactorial b=new NonStaticusingFactorial();

System.***out***.println(b.factorial(n));

}

}

**OUTPUT:**

**5**

**120**