short -signed(+ve,-ve) by default

unsigned(+ve) unsigned short

short 2byte 1byte=8bit

2byte=16bit

base 2

216 =65536

unsigned short= 0 to 65535 -10

signed=65536/2 - 32768 to +32768 2300

32768 ….,-3-2,-1 0,1,2,……….. 32767

232

Control statement(decision making):

if

if-else

else if

switch case

loop(for,while,do while,for each)

find out max between two digit

x=20,y=10

if()

code

else

code

operator:

1.Arithmatic

2.relational

3.increment & decrement

4.logical operator

5.bitwise

6.conditional

7.assignment operator

greater than > 20 >10 t 10>20f

greater than and equal to >= 30>=10t 30>=30t 40>=50f

equal to == 60==60t 50==60f

less than < 10<20t 34<10f

less than and equal to <= 10<=10t 49<=10f

not equal != 4!=5t 5!=5f

find out maximum between two numbers

Enter two numbers:

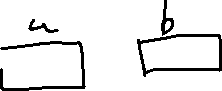
20

13

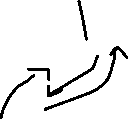
20 is maximum

int a,b;

sout(“Enter two digits”);



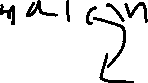
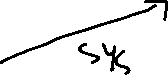
Scanner sc=new Scanner(System.in);



a=sc.nextInt();



b=sc.nextInt();



if(a>b)

sout(a+” is greater”);

else

sout(b+”is greater”);

Logical operator:

&&= logical AND

||=logical OR

! =logical NOT if( ( )&&( ) )

|  |  |  |
| --- | --- | --- |
| X | Y | X&&Y |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

1.wap to calculate increment appraisal of an any employee(HR). Increment % will be based on Goals:

Goals: I.60% should achieve in recruiting candidates(total >=70 candidtes)-the increment will be 20% of an basic salary. (bs\*20)/100

II.30% should achieve in recruiting candidates(total >=40 <70 candidtes)-the increment will be 15% of an basic salary.

III.10 should achieve in recruiting candidates(total >=10 <40candidtes)-the increment will be 10% of an basic salary.

Input of Basic salary and no.of candidates recruits will be taken from user.

Logical Or

|  |  |  |
| --- | --- | --- |
| X | Y | X||Y |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

b

0 1 2 3 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| b |  |  |  |  |

(bs\*20)/100;

Increment(++) and decrement(--)-(pre,post)

pre increment:

int x=10;

int y=++x; //1+x 1+10=11

sout(x); //11

sout(y);//11

post increment:

int x=10;

int y=x++; //y=10 x+1=10+1=11

sout(x); //11

sout(y);//10

pre decrement:

int x=10;

int y=--x; //1-x 1-10=9

sout(x); //9

sout(y);//9

post decrement:

int x=10;

int y=x--; //y=10 x-1=10-1=9

sout(x); //9

sout(y);//10

Loop-

for( initialization ; condition checking ; increment/decrement )

{

body of loop;

Dry run:

i=1<=5t

i=1+1=2<=5t

i=2+1=3<=5t

i=3+1=4<=5t

i=4+1=5<=5t

i=5+1=6<=5f

}

int i;

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

{

sysout(i);

}

o/p-

1

2

3

4

5

int i=1;

for(;i<=5;)

{

sysout(i);

i++;

}

write a program to count no of digits:

input:1234 o/p-no.of digits:4

num count

4

0

1234>0t

123>0t

(n!=0)

for(;num>0;) //1234>0 123>0 12>0 1>0 0>0f

{

count++; //1

num=num/10; //123

}

1.wap to check whether the number is perfect or not?

6

it is a positive integer which is equal to the sum of its positive divisors.

1 + 2 + 3=6

1 2 3 4 5 <6 n=6 sum=6

for(i=1;i<=n/2;i++)

{

if(n%i==0) //6%5 1==0f

sum=sum+i; //sum+=i; //3+3=6

}

if(n==sum)

sout(n+ “is a perfect number”);

else

sout(n+“ is not a perfect number”);

Dry run:

i=1<6t

i=2<6t

i=3<6t 3<=3

i=4<6t

i=5<6t

i=6<6f

3.wap to find out first and last digit of any number;

input=1234 first digit=1 last digit=4

nested for loop:

int I,j;

i=1<=3t

j=1<3t

j=2<3t

j=3<3f

i=2<=3t

j=1<3t

j=2<3t

j=3<3f

for(i=1;i<=3;i++) //external/outer/row

{

for(j=1;j<3;j++) //internal/inner/column

{

sout(j);

}

}

o/p:

i=1 i=2 i=3

j=1 1 j=1 1 j=1 1

j=2 2 j=2 2 j=2 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \* |  |  |  |  |
| \* | \* |  |  |  |
| \* | \* | \* |  |  |
| \* | \* | \* | \* |  |
|  |  |  |  |  |

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

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

i=4<=5t

j=1<=4t 2 3 4

i=1<=5t

j=1<=1t

j=2<=1f

i=2<=5t

j=1<=2t

j=2<=2t

j=3<=2f

i=3<=5t

{

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

{

sysout(“\*”);

}

sysout(“”);

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \_ | \_ | \_ | \_ | \* |
| \_ | \_ | \_ | \* | \* |
| \_ | \_ | \* | \* | \* |
| \_ | \* | \* | \* | \* |
| \* | \* | \* | \* | \* |

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

i=1<=5t

j=5>1t 4>1t 3>1t 2>1t 1>1f

k=1<=1t 2<=1f

i=2<=5t

j=5>2t 4>2t 3>2t 2>2f

k=1<=2t 2<=2t k=3<=2f

for(i=1;i<=5;i++)//row

{

for(j=5;j>i;j--) //space

{

sout(“ ”);

}

for(k=1;k<=i;k++)//col

{

sout(“\*”);

}

sysout();

}

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

while loop-

int i=1;

while(i<5)

{

sout(“hii”);

i++;

}

hi hi hi hi

Strong number:

145

1!+4!+5!= 145 1+ 24+ 120

return type method\_name(argument list)

{

body of method

}

num=145 fact I rem sum temp

145

145

1

1

0

temp=num;

while(num>0) //145>0 t 14>0t 1>0t 0>0f

{

rem=num%10; //1

fact=1;

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

{

fact\*=i; //fact=fact\*i

}

sum+=fact; //sum=sum+fact 144+1=145 sum=145

num=num/10; //num/=10; //0

}

}//while

if(sum==temp)

{

sout(“strong number”);

}

else

sout(“not a strong number”);

class Student

{

78967

id=101

name=gargy

address=kolkata

int id;

String name;

String address;

5634

id=

name=sushri sardar

address=null

void persueDegree()

{

}

}

Student gargy=new Student();

Student sushri=new Student();

|  |
| --- |
| Sushri  5634 |
| gargy  78967 |

gargy.name =gargy 78967.id =50

sushri.name=sushri sardar

gargy.setData(101,”gargi”,”kolkata”);

class Student

{

int id;

String name;

String address;

void setData(int id,String name,String address)

{

this.id=id; //1289.id=102

this.name=name; //1289.name=chandrima

this.address=address; //1289.address=saltlake

}

}

this keyword:

variable

method

constructor

this is a reference variable created by compiler.

35890

id=101

name=sruti

address=kolkata

|  |
| --- |
|  |
| 6789 this  1289 |

Student s1=new Student();

1289

id=101

name=sruti

address=saltlaje

s1.setData(101,”sruti”,”kolkata”); name=s1.name

|  |
| --- |
|  |
| S2  1289 |
| S1  35890 |

Student s2=new Student();  
s2.setData(102,”chandrima”,”saltlake”);

count=2

institute

Anudip foundation

class Student

{

int id;

String name;

static int count=0;

public Student(int id,String name)

{

this.id=id;

this.name=name;

count++;

sysout(count); //1 2

}

public Student(Student s)

{

id=s.id;

name=s.name;

}

}

7890

id=101

name=ronita

6789

id=101

name=ronita

count=1

count=2

Student s1=new Student(101,”ronita”);

Student s2=new Student(s1);

|  |
| --- |
| S2  6789 |
| S1  7890 |

1.single inheritance

User

name

display(){name}

parent/super/base

extends tightly coupled inheritance- IS-A relationship

Student

marks

calculatingTotalmarks(){}

child/sub/derived

2.multilevel inheritance

c

B

A

grand parent

intermediate class

child

3.hierarchical inheritance

Bank

rateOfInterest();

BOI

rateOfInterest(){7.25%}

Axis

rateOfInterest(){10%}

SBI

rateOfInterest(){6.75}

1.Override

2.abstract method

Runtime polymorphism: overriding

1.Is -A relationship

2.same method name

3.same parameter list

class User

User

7890

id=101

name=sananda

{

int id;

String name;

public User(int id,String name)

{

person

4567

salary=78000.0

this.id=id;

this.name=name;

}

}

class Person extends User{

public Person(int id,String name,float salary)

{

super(id,name);

this.salary=salary;

}

}

Person p=new Person(101,”sananda”,78000.0f);

|  |
| --- |
| Super  7890 |
| This  4567 |

1.multiple:

C

B

A

2.hybrid

C

D

B

A

D

C

B

A

class User

{

int id;

display();

}

class Person

{

String name;

Person(String name){this.name=name;}

}

class student extends User,Person

{ String marks;

student(String name,String marks)

{

super(name);

}

set();

get();

}

main()

{

Student s=new Student();

}

|  |
| --- |
| Super  9023 |
| This  7845 |

A

fun();

interface B extends A interface C extends A

C

int fun();

B

void fun();

D

fun(){“D class”}

class D implements B,C

Array:

int a=67;

a=56

Array is a collection of homogeneous elements which has contiguous memory location.

int arr[]={12,34,56};

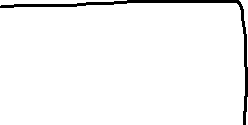
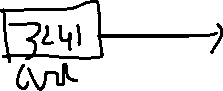
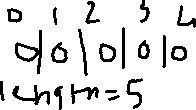
int arr[]=new int[5];



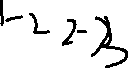
datatype[] arr;

datatype []arr;

datatype arr[]’



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

0 1 2 3



0 1 2 3 4 5 6 7 8 9

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 56 | 23 | 6 | 89 | 1 | 63 | 84 | 90 | 36 | 29 |

key=84 ,int pos=-1;

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

{

if(arr[i]==key) //84==84

{

pos=i; //pos=6

break;

}

}

if(pos!=-1)

sout(“element found at” +(pos+1))

else

sout(“data not found”);

Dry run:

i=0<10t

i=1<10

2 3 4 5 6

89

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 6 | 23 | 56 | 89 |

0 1 2 3 4 temp

i=0<5 t j=0+1=1<5 t j=2<5t j=3<5 j=4<5

j=5<5f

i=1<5 j=1+1=2<5 j=3 j=4<5 j=5<5f

i=2<5t j=2+1=3 j=4

i=3<5t j=3+1=4 j=5<5f

i=4<5

I j

for(i=0;i<size-1;i++)

{

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

{

if(arr[i]>arr[j]) //arr[3]>arr[4] 89>56t

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

Insert element in a specific position?

0 1 2 3 4 5 6 7 8 9



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 78 | 5 | 6 | 7 | 9 |  |  |  |  |

0 1 2 3 4 5 6

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 78 | 5 | 100 | 6 | 7 | 9 |  |  |  |

int[] arr=new int[10];

size=~~5 6~~ 7

pos=4 key=100 arr[5]=arr[4]

for(i=0;i<size;i++) 0 1 2 3 4 5

i=5>=2

arr[5]=arr[5-1] arr[5]=arr[4]

i=4>=2

arr[4]=arr[4-1] arr[4]=arr[3];

i=3>=2

arr[3]=arr[3-1] arr[3]=arr[2];

i=2>=2

arr[2]=arr[2-1] arr[2]=arr[1];

i=1>=2f

for(int i=size; i>=pos ;i--) //i=3>=4

{

arr[i]=arr[i-1]; //arr[4]=arr[3]

}

arr[pos-1]=key; //arr[4-1] arr[3]=100

size++;size+1 6+1 7

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 7 | 9 | 13 | 31 | 33 | 41 | 52 | 59 | 99 |

0 1 2 3 4 5 6 7 8 9 start end

key=33

start=0

end=length-1; //9

mid=(start+end)/2 (0+9)/2=4 mid=4

33>arr[mid] //arr[mid]<33

33>arr[4]

33>31 start=mid+1 4+1=5 start=5

5 6 7 8 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 33 | 41 | 52 | 59 | 99 |

start end

mid=(start+end)/2 (5+9)/2 mid=7

33>arr[mid]

33>arr[7]

33>52 end=mid-1 end=6

5 6

|  |  |
| --- | --- |
| 33 | 41 |

start end

mid=(start+end)/2 (5+6)/2 mid=5

33==arr[mid]

33==arr[5]

33==33 t search successful

start=5

end=4

if start index is > end so search unsuccessful

Array of objects:

Class Student

{

int id;

String name;

String address;

//setter getter

//cons

tostring

}

5678

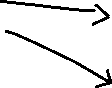
Student obj1=new student(1,”shrabani”,”Kolkata”);



Student obj2=new student(2,”ankana”,”Kolkata”);

1245

Student obj3=new student(3,”keshri”,”Kolkata”);



9076

Student[] arr=new Student[5];

arr[0]=obj1;

arr[1]=obj2;

arr[2]=obj3;

0 1 2 3 4 5 6 7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 5678 | 1235 | 9076 |  |  |  |  |  |

for(int i=0;i<3;i++)

{

sout(arr[i]);//5678 1245 arr[i].getId()

}