

indentation is for us programmer
compiler don't need it.

even number series?

class even {

public static void main(String[] args) {

length i, n=10;

for (i=2; i<=n; i+=2) {
 System.out.println(i);

 // addition method

}

 // odd numbers

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

 if (i%2 != 0)

 System.out.println(i)

 // odd

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

 if (i%2 == 0)

 System.out.println(i)

 // division method.

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

 System.out.println(i*2);

 // multiplication method

- write down program to print even number upto ~~at~~ n.
- write down program to print n even/odd numbers.

class even {

public static void main(String s[]) {

~~length~~ int start, end, n=10;

 for (i=start; i<=end; i++) {

 if (i%2 == 0) {

 System.out.println(i);

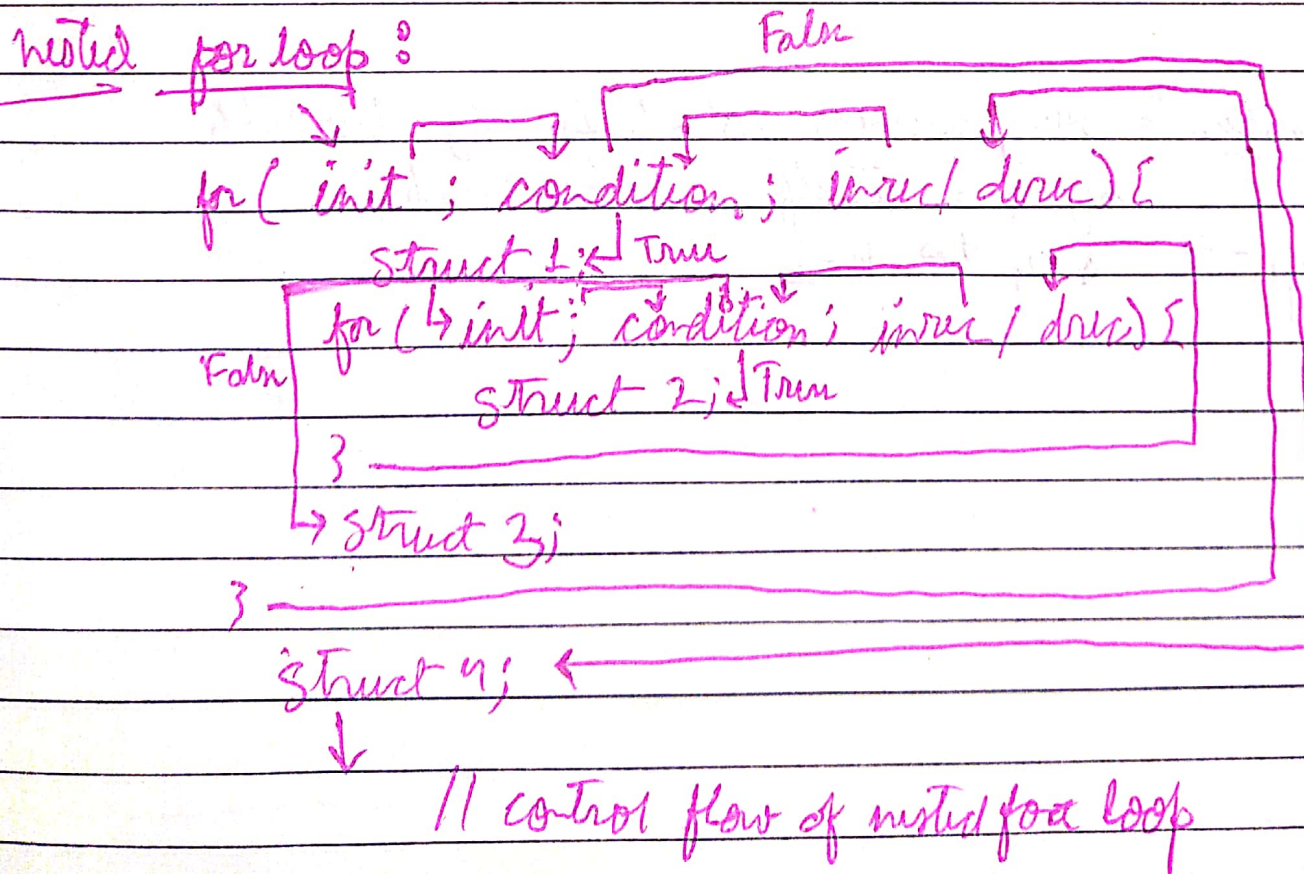
 }

 }

for odd numbers
 $2*i-1$

```
class even {
    public static void main (String s[]) {
        byte i, n = 5;
        for (i=1; i<=n; i++) {
            System.out.println(2*i);
        }
    }
} // Program for n even/odd numbers.
```

n = 5	
i = 1	$2*1 = 2$
i = 2	$2*2 = 4$
i = 3	$2*3 = 6$
i = 4	$2*4 = 8$
i = 5	$2*5 = 10$
i = 6	




```

class Pat1 {
    public static void main (String s[]) {
        int n=5;
        for (int i=1; i<=n; i++) {
            for (int j=1; j<=i; j++) {
                System.out.print(j);
            }
            System.out.println("");
        }
    }
}

```

1
 1 2
 1 2 3
 1 2 3 4
 1 2 3 4 5

Prime Number : Any number which is only divisible
 by 1 or itself is prime number
 loop \rightarrow 2 to $n-1$

Class Prime {

public static void main (String args[]) {

int i, k, n=10;

for (i=2; i<=n; i++) {

for (k=2; k<=i/2; k++) {

if (i%k==0)

k>i/2 break;

if (~~i/2~~) {

System.out.println(i);

}

}

}

n=10

i=2

~~i=2~~

k=2

i=3

k=2,3

i=4

k=2

i=5

k=2,3,4,5

i=6

k=2

i=7

k=2,3,4,5,6,7

i=8

k=2

i=9

k=2,3

i=10

k=2

i=11

for optimized dry run:

$i=2, t=1$

$K=2$

$i=3, t=1$

$K=2$

$i=4, t=2$

$K=2$

$i=5, t=2$

$K=2, 3$

$i=6, t=3$

$K=2$

$i=7, t=3$

$K=2, 3, 4$

Logic of Optimized solution:

for ($i=2; i \leq n; i++$) {
 $t = i/2$;

 for ($K=2; K \leq t; K++$) {
 if ($i \% K == 0$)
 break;

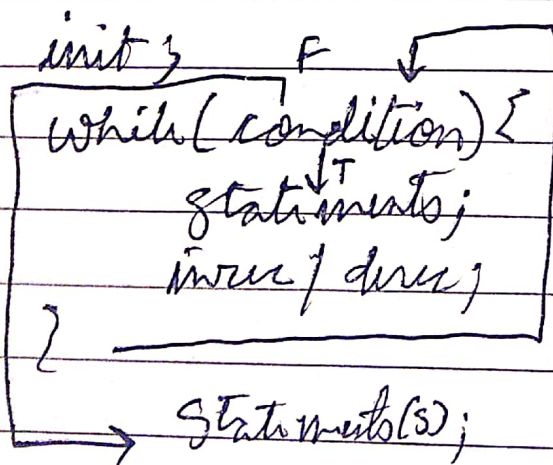
 }

 if ($K > t$)

 system.out.println(i);

}

While loop:



```

class WhileDemo {
    public static void main(String args[]) {
        int i, n=10;
        i=1;
        while(i<=n) {
            if(i%2==0)
                System.out.println(i);
            i++;
        }
    }
}

```

Exit checking loop:

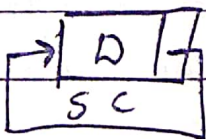
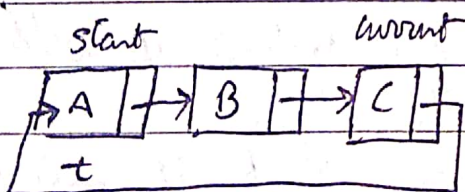
```

do {
    stats;
    inc/dec;
} while(condition);

```

we used it in circular linked list Traversal:

eg.



this will
not work for single
node circular
linked list

```

t = S
while(t != S)
{
    Print t -> info;
    t = t -> link;
}

```

```

t = S
do {
    Print t -> info;
    t = t -> link;
} while(t != S);

```

this will work for
single linked list.

Class Demo 1

```
public static void main (String s[]) {  
    int i, n=10;  
    i=1;  
    do {  
        if ( i%2 != 0 ) {  
            System.out.println(i);  
        }  
        i++;  
    } while ( i <= n );  
}
```

★ Access Modifiers :

Private

Public

Abstract