ITERATION/REPITITION STATEMENTS

ITERATIVE/LOOPING STATEMENTS

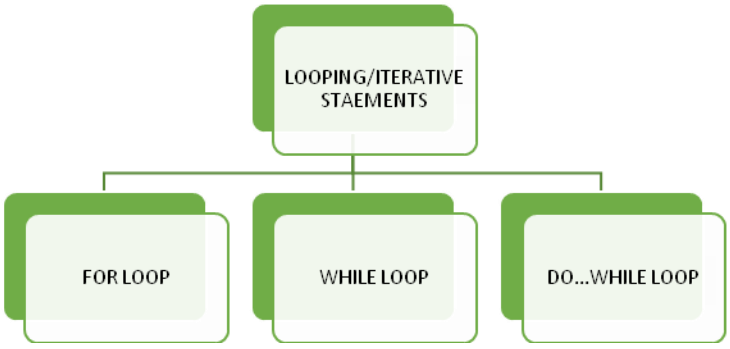
An iterative or loop statement allows us to execute a statement or group of statements multiple times until the condition is true. It stops when the condition becomes false.

NEED OF ITERATIVE/LOOPING STATEMENTS

- There may be a situation, when you need to execute a block of code several number of times.
- For example, if you are asked to display your name 1000 times on the screen using a program.
- It is not worth writing the same display statement 1000 times in the program.
- It may lead to errors and makes the program code unnecessarily lengthy.
- To avoid all these loops are essential.

TYPES OF ITERATIVE STATEMENTS

There are three types of iterative/looping statements.



for LOOP STATEMENT

SYNTAX:

for(initialization; test exp; increment/decrement exp)

{

statements;

}

There are five major elements in a loop:

a) **Control/loop variable**: Every loop has one control variable which keeps track of number of times the loop executes.

b) **Initialization expression**: It assigns initial/starting value to the loop variable. It is executed only once in the beginning of the loop.

c) **Test Expression**: The test expression decides whether the loop will continue to execute or not. If the test condition is true, the loop body gets executed otherwise the loop is terminated. It is checked every time before entering in the body of the loop.

d) **Update (Increment/Decrement) Expression**: It changes the value of the loop variable. The increment/decrement expression is executed every time after the body of the loop is executed.

e) **The Body of the loop**: The statements, which are executed repeatedly till the test expression evaluates to false form the body of the loop.

for LOOP EXAMPLE

Q Write a loop statement using for loop to display "HELLO" five times on the screen.

```
for(int i=1; i<=5; i=i+1)
{
System.out.println("HELLO");
}
```

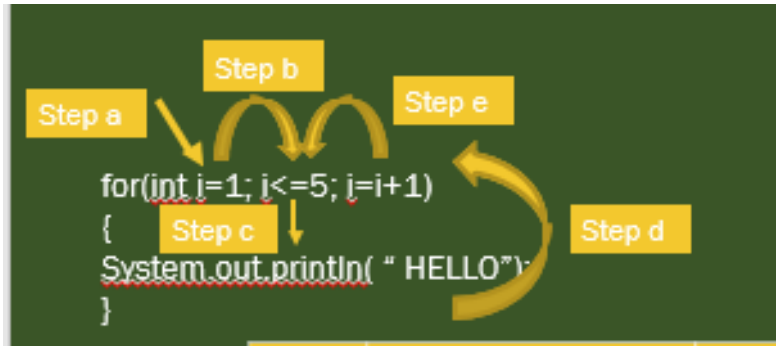
Loop variable	i
Initialization expression	i=1
Test expression	i<=5
Update expression	i=i+1

Working of the loop:

1. The initialization expression is executed and variable i is initialized by value 1.
2. Then the test expression is executed and it is checked whether the test expression is true or false.
3. If the test expression is true, the body of the loop is executed ie HELLO is printed on the screen.

- ❖ After executing the body of the loop, program control jumps to the update expression and the value of i is incremented by 1.
- ❖ After the update expression the program control jumps back to the test expression and test expression is evaluated for true or false value.
- ❖ If the value of test expression is true again all the steps from step 3 are executed.

4. If the value of the test expression is false, the loop will stop executing and it terminates. In this program the test expression evaluates to false when the value i becomes greater than 5 (ie i=6)



NOTE: Step *a* and *b* are executed only once in the beginning of the loop ie when the loop starts. Steps *c, d, e* are executed repeatedly until the test expression becomes false.

S.No.	Value of loop variable	Test expression	Result of execution of body of loop	Update expression
1	i=1	1<=5 ,TRUE	HELLO printed	i=i+1
2	i=2	2<5, TRUE	HELLO printed	i=i+1
3	i=3	3<5, TRUE	HELLO printed	i=i+1

4	**i=4**	**4<5, TRUE**	**HELLO printed**	**i=i+1**
5	**i=5**	**5=5, TRUE**	**HELLO printed**	**i=i+1**
6	**i=6**	**6>5, FALSE**	**LOOP TERMINATED**	

for LOOP: SOME MORE EXAMPLES

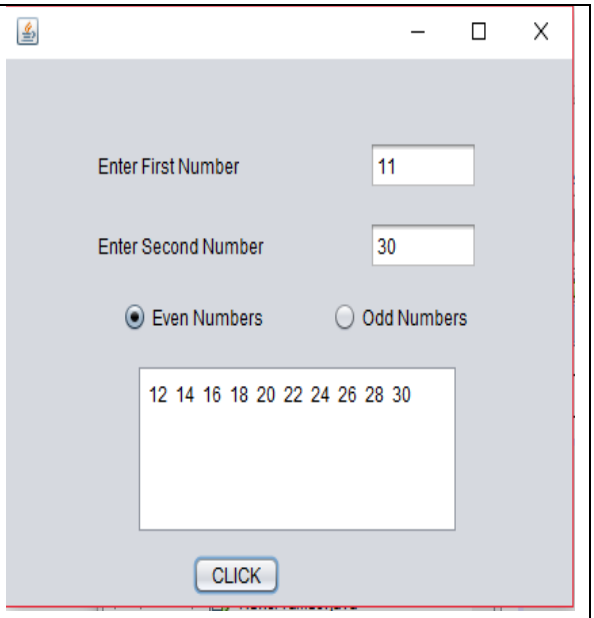
EXAMPLE	EXPLANATION
for(int k=1; k<10;k++) System.out.println(" "+ k);	The loop will execute 10 times and will display numbers from 1 to 10.
for(int m=100; m>0; m=m-25) System.out.println(" "+ m);	The loop will execute 5 times and will display numbers from 100, 75, 50, 25. Loop variable will be decremented by 25 at every step.
for(int m=100; m>0; m=m +25) System.out.println(" "+ m);	The loop will execute infinitely as value of m is incremented by 25 at every step. It will always be greater than zero and test expression will never become false.
for(int k=1; k<10;k++); System.out.println(" "+ k);	In this case there is a semicolon after the loop statement. As a result, loop will never reach its body. It will execute ten times from 1 to 10 and evaluates to false when the value of k becomes 11. After this the system.out.println() command is executed and 11(the last value of k) is printed on the screen.
for(int k=1; k>10;k++) System.out.println(" "+ k);	The loop will not execute at all because the test expression is false in the beginning itself. There will be no output.

EXAMPLE OF FOR LOOP

Q WAP to input starting number and ending number and display either even or odd number between these two numbers depending upon user's choice.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {

    int a= Integer.parseInt(jTextField1.getText());
    int b= Integer.parseInt(jTextField2.getText());
    if(jRadioButton1.isSelected())
    {  for(int i=a; i<=b;i++)
       if(i%2 == 0)
       jTextArea1.append(""+ i + "  ");
    }
    if(jRadioButton2.isSelected())
    {  for(int i=a; i<=b;i++)
       if(i%2 != 0)
       jTextArea1.append(""+ i + "  ");
    }
  }
```



Click to view video on for loop

while STATEMENT

SYNTAX:

while(test expression)

{

loop body

}

- Initialization expression is outside the loop.
- Test expression is checked before entering the loop. If the test expression evaluates to false, loop will not be executed.
- Update expression is present in the loop body itself.

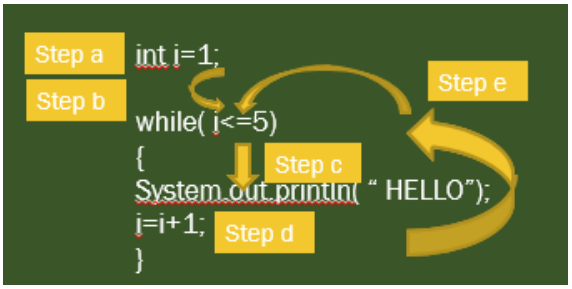
Q Write a loop statement using while loop to display "HELLO" five times on the screen.

```
int i=1;
while( i<=5)
{System.out.println( " HELLO");
i=i+1;
}
```

Working of the loop:

1. The initialization expression is executed and variable i is initialized by value 1.
2. Then the test expression is executed and it is checked whether the test expression is true or false.
3. If the test expression is true, the body of the loop is executed ie HELLO is printed on the screen.
   - ❖ After executing the body of the loop, program control jumps to the update expression and the value of i is incremented by 1.
   - ❖ After the update expression the program control jumps back to the test expression and test expression is evaluated for true or false value.
   - ❖ If the value of test expression is true again all the steps from step 3 are executed.
4. If the value of the test expression is false, the loop will stop executing and it terminates. In this program the test expression evaluates to false when the value i becomes greater than 5(ie i=6)

while LOOP EXAMPLE EXPLAINED



Loop variable	i
Initialization expression	i=1
Test expression	i<=5
Update expression	i=i+1

NOTE: Step *a* and *b* are executed only once in the beginning of the loop ie when the loop starts. Steps *c,d,e* are executed repeatedly until the test expression becomes false.

S.No.	Value of loop variable	Test expression	Result of execution of body of loop	Update expression
1	i=1	1<=5 ,TRUE	HELLO printed	i=i+1
2	i=2	2<5, TRUE	HELLO printed	i=i+1
3	i=3	3<5, TRUE	HELLO printed	i=i+1
4	i=4	4<5, TRUE	HELLO printed	i=i+1
5	i=5	5=5, TRUE	HELLO printed	i=i+1
6	i=6	6>5, FALSE	LOOP TERMINATED	

while LOOP: SOME MORE EXAMPLES

EXAMPLE	EXPLANATION

int k=1;
while(k<10)
{System.out.println(" "+ k);
	The loop will execute 10 times and will display numbers from 1 to 10.

k++;}	

int m=100;
while(m>0)
{System.out.println(" "+ m); m=m-25;}
	The loop will execute 5 times and will display numbers from 100, 75, 50,25. Loop variable will be decremented by 25 at every step.

int m=100;
while(m>0)
{System.out.println(" "+ m);
m=m +25;}
	The loop will execute infinitely as value of m is incremented by 25 at every step. It will always be greater than zero and test expression will never become false.

int k=1;
while(k<10);
{System.out.println(" "+ k);
k++);
	In this case there is a semicolon after the loop statement. As a result, loop will never reach its body. It will execute ten times from 1 to 10 and evaluates to false when the value of k becomes 11. After this the system.out.println() command is executed and 11(the value of k) is printed on the screen.

int k=1;
while(k<10);
{System.out.println(" "+ k);
k++);
	The loop will not execute at all because the test expression is false in the beginning itself. There will be no output.

int m=100;
while(0)
{System.out.println(" "+ m); m=m-25;}
	The loop will not execute at all because the test expression has value 0 which is considered as false in the beginning itself. There will be no output.

int m=100; while(1)	The loop will execute infinitely because the test expression has value 1 which is considered as true.

{System.out.println(" "+ m); m=m-25;}	The loop will execute indefinitely.

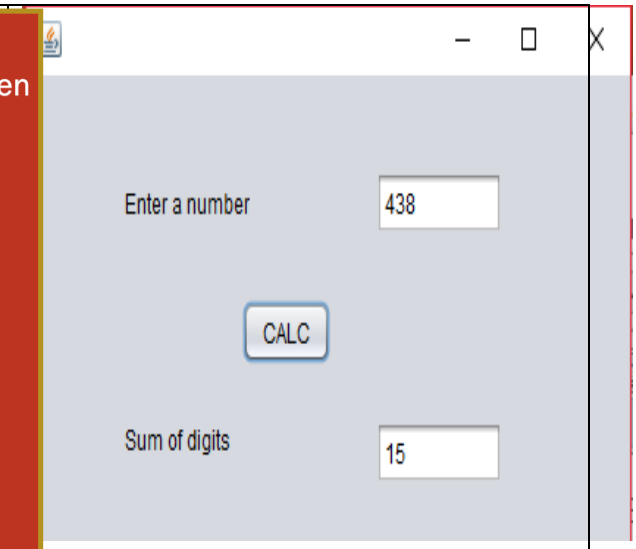
char val=65;
while(val<70)
{System.out.println(" "+val); val++);
	In this case the variable val will take 'A' (ASCII code of A is 65) as its value. It will display the alphabets A,B,C,D,E on the screen.

Click to view video on while and do..while loop

EXAMPLE OF WHILE LOOP

Q WAP to input a number and display the sum of its digits.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEven
t evt) {
        int num =
Integer.parseInt(jTextField1.getText());
        int a=0, sum=0;
        while(num>0)
        {
           a=num%10;
           sum = sum + a;
           num = num/10;
        }
        jTextField2.setText(""+ sum);
                                    }
```



do while STATEMENT

SYNTAX:

do

{

loop body

} while(test expression);

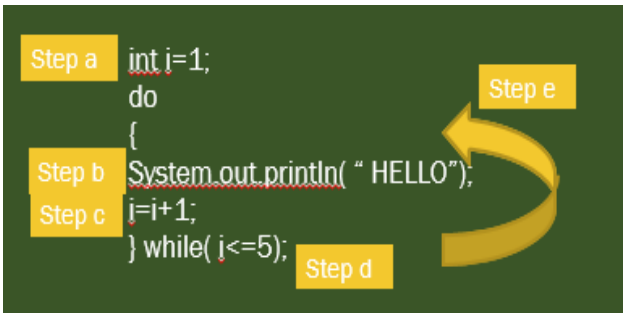
- Initialization expression is outside the loop.
- Test expression is checked after the loop body is executed.
- Even if the test expression is wrong, loop will be executed at least once because the test expression is checked after the loop body is executed.
- Update expression is present in the loop body itself.

Q Write a loop statement using do...while loop to display "HELLO" five times on the screen.

```
int i=1;
do {
System.out.println( " HELLO");
i=i+1;
} while( i<=5);
```

Working of the loop:

1. The initialization expression is executed and variable i is initialized by value 1.
2. Then the body of the loop is executed ie HELLO is printed on the screen.
    - After executing the body of the loop, program control jumps to the update expression and the value of i is incremented by 1.
    - After the update expression the program control jumps to the test expression and test expression is evaluated for true or false value.
    - If the value of test expression is true again all the steps from step 2 are executed.
3. If the value of the test expression is false the loop will stop executing and it terminates. In this program the test expression evaluates to false when the value i becomes greater than 5(ie i=6)



Loop variable	i
Initialization expression	i=1
Test expression	i<=5
Update expression	i=i+1

NOTE: Step *a* is executed only once in the beginning of the loop ie when the loop starts. Steps *b,c,d,e* are executed repeatedly until the test expression becomes false.

S.No.	Value of loop variable	Test expression	Result of execution of body of loop	Update expression
1	i=1		HELLO printed	i=i+1
2	i=2	2<5, TRUE	HELLO printed	i=i+1
3	i=3	3<5, TRUE	HELLO printed	i=i+1
4	i=4	4<5, TRUE	HELLO printed	i=i+1
5	i=5	5=5, TRUE	HELLO printed	i=i+1
6	i=6	6>5, FALSE	LOOP TERMINATED	

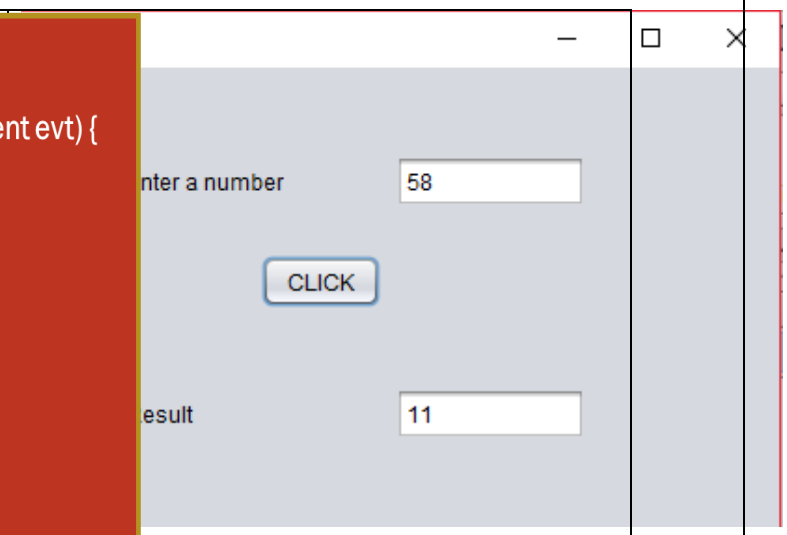
do while LOOP: SOME MORE EXAMPLES

EXAMPLE	EXPLANATION
int k=1; do {System.out.println(" "+ k); k++; }while(k<10);	The loop will execute 10 times and will display numbers from 1 to 10.
int m=100; do {System.out.println(" "+ m); m=m-25; }while(m>0);	The loop will execute 5 times and will display numbers from 100, 75, 50,25. Loop variable will be decremented by 25 at every step.
int m=100; do {System.out.println(" "+ m); m=m +25; } while(m>0) ;	The loop will execute infinitely as value of m is incremented by 25 at every step. It will always be greater than zero and test expression will never become false.
int k=1; do {System.out.println(" "+ k); k++; }while (k>10);	Although the test condition is false the loop is executed once because the test condition is checked after the execution of the body of the loop. The output will be 1.

EXAMPLE OF DO…WHILE LOOP

Q WAP to input a number and find the sum of all numbers divisible by 5 from that number till 0.

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    int a = Integer.parseInt(jTextField1.getText());
    int count = 0;
    do
    {  if(a%5 == 0)
        count++;
       a--;
    }while(a>0);
    jTextField2.setText("" + count);
  }
```



ENTRY CONTROLLED AND EXIT CONTROLLED LOOPS

ENTRY CONTROLLED LOOP	EXIT CONTROLLED LOOP
1. Test condition is checked in the beginning, before entering the loop.	1. Test condition is checked at the end, after the body of the loop is executed.
2. The loop will not be executed if the test condition is false in the beginning of the loop.	2. The loop will be executed at least once even if the test condition is false in the beginning of the loop.
3. Eg. for loop, while loop	3. Eg. do.. while loop

COMPARING FOR, WHILE AND DO….WHILE STATEMENTS

FOR LOOP	WHILE LOOP	DO…WHILE LOOP

1. The test condition is checked before entering the loop.	1. The test condition is checked before entering the loop.	1. The test condition is checked after the loop body is executed.
2. It is also called entry controlled loop.	2. It is also called entry controlled loop.	2. It is also called exit controlled loop.
3. The loop will not be executed if the test condition is false in the beginning of the loop.	3. The loop will not be executed if the test condition is false in the beginning of the loop.	3. The loop will be executed at least once even if the test condition is false in the beginning of the loop.

Click to view video on Break and Continue