FAST

National University of Computer and Emerging Sciences Peshawar

Lecture # 06

Software Construction and Development (Java Programming)

Instructor: Muhammad Abdullah Orakzai

DEPARTMENT OF COMPUTER SCIENCE

الذی علم بالقلم۔ علم الانسان ما لم یعلم۔

Loops in Java

Contents

Loops

In computer programming, a **loop** is a sequence of instructions that is continually repeated until a certain condition is reached.

A **loop** statement allows us to execute a statement or group of statements multiple times

Types of Loop

1) for loop
2) while loop
3) do while loop
4) for-each loop(Enhanced For Loop)

Loops

Pretested Loop: Loop in which condition is checked first.

e.g. for loop and while loop

Post tested Loop: Loop in which condition is checked at the end.

e.g. do while loop

Determined loop/Definite loop: Loop for fixed repetition

e.g. for loop

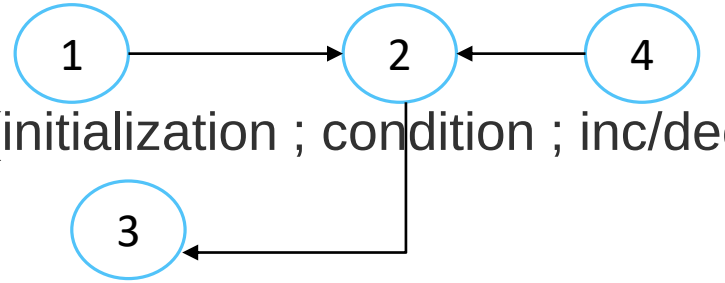
Undetermined loop/Indefinite loop: loop not for fixed repetition

E.g. while loop and do while loop

1) for loop

- for loop is used to a statement or group of statement for a fixed number of time.
- If the number of iteration is fixed then it is recommended to use for loop.

Syntax:




```
for(initialization ; condition ; inc/dec)
{
// Statement(s)
}
```

1) for loop…

Order of Steps in for loop:

1) 1st initialization is performed.
2) Secondly condition is checked
3) In 3rd step statement is executed means control goes to body of the loop.
4) In 4th step incrementation or decrementation is performed.
5) Again condition is checked an so on.

Note: In loops variable is called counter variable

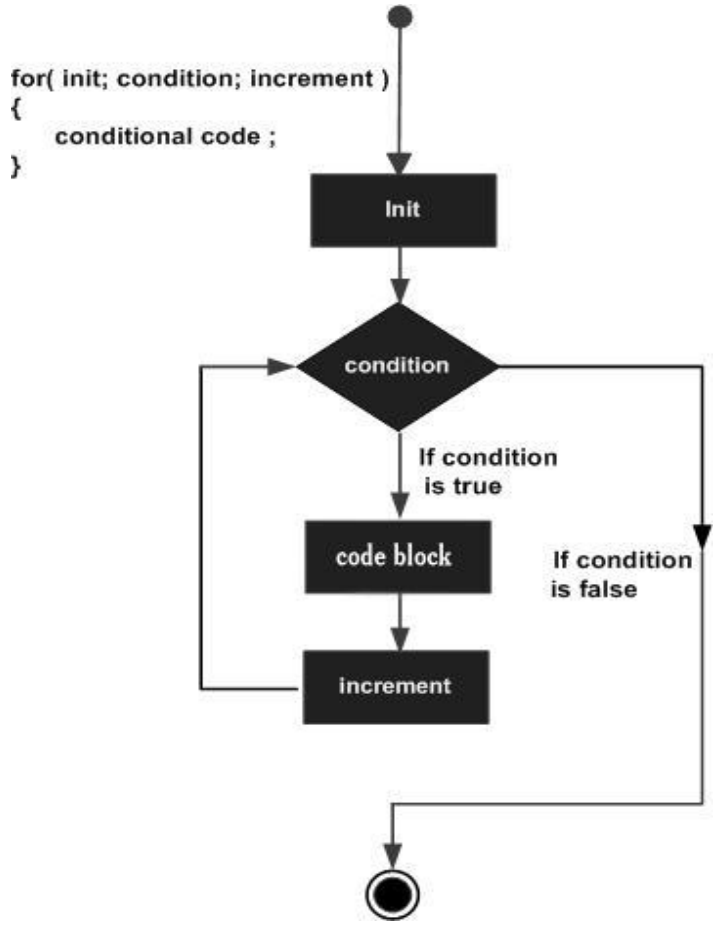
i= i+1; OR i+=1;

for loop Example

```
public class Test {

   public static void main(String args[]) {

      for(int x = 10; x < 20; x = x + 1) {
         System.out.print("value of x : " + x );
         System.out.print("\n");
      }
   }
}
```



for loop Example

```
class ForDemo
{
public static void main(String args[])
 {
int i;
for(i=0;  i<=10 ; i++)
{      System.out.println(“Kmayab Jawan Program”);   }
}
}
```

Java infinitive for loop

If you use two semicolons (; ;) in the for loop it will be infinitive for loop.

Syntax:

```
for(; ;)
{
Statement(s);
}
```

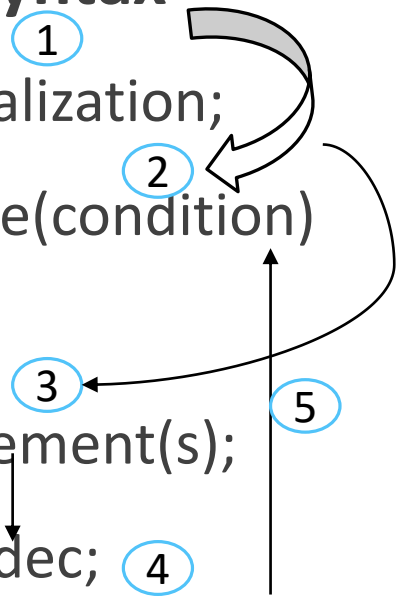
for loop Tasks

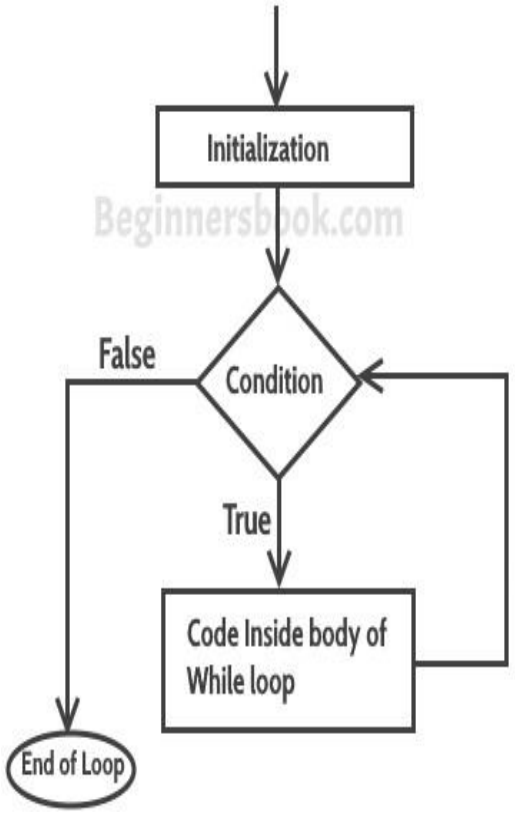
1. Write a java program which display first 10 number using for loop.
2. Write a java program which display even and odd number using for loop.
3. Take a number from user and make a table of that number using for loop.
4. Take a number from user and find factorial of that number using for loop.

2) while loop

- Is used when number of iteration is not fixed.
- **Syntax**

```
initialization;
while(condition)
{
statement(s);
inc/dec;
}
```





while loop Example

```
class WhileLoopExample {

public static void main(String args[])   {

int i=10;

while(i>1)
{
 System.out.println(i);
     i--  ;
}
  }
}
```

while loop Example

```
class WhileDemo
{
public static void main(String args[])    {
int i=1;
while(i<=10)
{
System.out.println(“Kmayab Jawan Program”);
i++;
}
} }
```

Java infinitive while loop

If you pass true or true value in the while loop, it will be infinitive while loop.

Syntax:

```
while(true)
{
   Statement(s);
 }
```

Java infinitive while loop…

Example

```
while(true)
{
System.out.println(“infinitive while loop”);
}
```

while loop Tasks

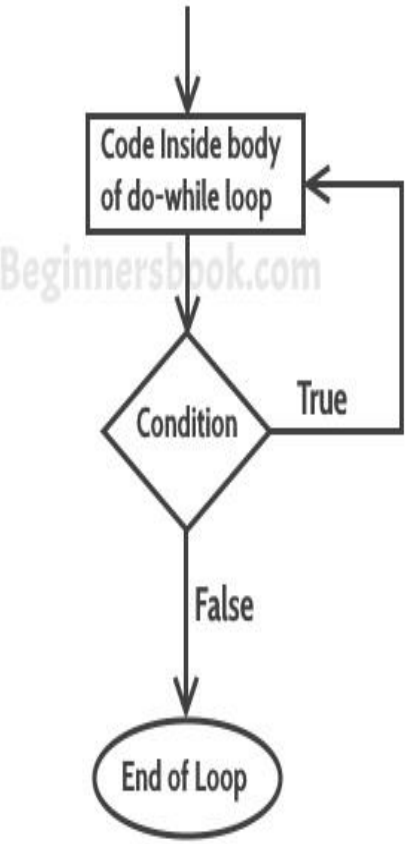
1. Write a java program which display first 10 number using while loop.
2. Write a java program which display even and odd number using while loop.
3. Take a number from user and make a table of that number using while loop.
4. Take a number from user and find factorial of that number using while loop.

3) do while loop

- An indefinite loop. Best used when the number of iteration is unknown.
- Used when you will execute the loop at least once.

Syntax

```
initialization;
do{
statement(s);
inc/dec;
}
while(condition) ;
```



do while loop Example

```
class DoWhileLoopExample {

public static void main(String args[]){

int i=10;     // intializaton

do
{
     System.out.println(i);

     i--  ;  //decrementation
}
while(i>1) ;
 }

}
```

do while loop Example

```
class DoWhileDemo
{
public static void main(String args[])    {
int i=1;
do
{
System.out.println(“Kmayab Jawan Program”);
i++;
}
while(i<=10);
} }
```

Java infinitive do while loop

If you pass true or true value in the do while loop, it will be infinitive do while loop.

Syntax:

```
do
{
Statement(s);
}
while(true);
```

Java infinitive do while loop…

Example

```
do
{

System.out.println(infinitive while loop);

}
while(true);
```

do while loop Tasks

1. Write a java program which display first 10 number using do while loop.
2. Write a java program which display even and odd number using while loop.
3. Take a number from user and make a table of that number using do while loop.
4. Take a number from user and find factorial of that number using do while loop.

Enhanced for loop (for-each loop)

- Works with array.
- Is used for traversing in array.
- It is easy to use than simple for loop because we do not need to increment or decrement counter variable.

Syntax

```
for (data type variableName : arrayName)
{    statement(s);   }
```

Data type must be same as that of array data type.

Enhanced for loop example

```
class ForEachExample1{
 public static void main(String args[]){
//declaring an array
 int arr[]={12,13,14,44};
//traversing the array with for-each loop
 for(int i:arr){
     System.out.println(i);
   } }
}
```

Break Statement

The break statement terminates the execution of the loop when it is used inside the body of the loop.

Syntax: break;

Break Statement Example

```
public class BreakExample1 {
   public static void main(String args[]){
      int num =0;
      while(num<=100)
      {
          System.out.println("Value of variable is: "+num);
          if (num==2)
          {
             break;
          }
          num++;
      }
      System.out.println("Out of while-loop");
   }
}
```

Output:

```
Value of variable is: 0
Value of variable is: 1
Value of variable is: 2
Out of while-loop
```

Break Statement Example

```
public class BreakExample3 {

    public static void main(String args[]){
          int num=2;

          switch (num)
          {
               case 1:
                   System.out.println("Case 1 ");
                   break;
               case 2:
                   System.out.println("Case 2 ");
                   break;
               case 3:
                   System.out.println("Case 3 ");
                   break;
               default:
                   System.out.println("Default ");
          }
    }
}
```

Output:

```
Case 2
```

Continue Statement

- The continue statement shifts the control back to the beginning of the loop.
- It is used inside the body of the loop.
- It is used to continue loop.
- It continues the current flow of the program and skips the remaining code at specified condition.

Syntax: continue;

Continue Statement Example: Use of continue in for loop

```
public class ContinueExample {

   public static void main(String args[]){
        for (int j=0; j<=6; j++)
        {
           if (j==4)
           {
              continue;
           }

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

Output:

```
0 1 2 3 5 6
```

Continue Statement Example: Use of continue in While loop

```
public class ContinueExample2 {

   public static void main(String args[]){
        int counter=10;
        while (counter >=0)
        {
           if (counter==7)
           {
                counter--;
                continue;
           }
           System.out.print(counter+" ");
           counter--;
        }
   }
}
```

Output:

```
10 9 8 6 5 4 3 2 1 0
```

Sentinel Condition

Truthfulness and falseness depends upon user input.

Sentinel Condition while loop…

```
Import java.util.Scanner;
Class AddNoWhile
{    // class body starts
public static void main(String args[])
{   // main method body starts
Scanner input = new Scanner(System.in)
int n1, n2;
String choice = “Yes”; // choice declaration and initialization
while(choice.equals(“Yes”))
{   // while body starts
```

Sentinel Condition while loop…

```
System.out.println(“Enter 1st no”);
n1=input.nextInt();
System.out.println(“Enter 2nd no”);
n2=input.nextInt();
System.out.println(“Sum=”+(n1+n2));
System.out.println(“Do you want to do more addition (Yes/No)”);
choice= input.next();
} // while loop body closed
System.out.println(“Thank you for using while loop”);
} // main method body closed
} // class body closed
```

Sentinel Condition while loop…

Note:

1) In the previous example all choices except “Yes” will be considered no and loop will be terminated.
2) In this while loop example we initialize string at top because condition is checked at the top.

Sentinel Condition do while loop

```
Import java.util.Scanner;
Class AddNoWhile
{     // class body starts
public static void main(String args[])
{    // main method body starts
Scanner input = new Scanner(System.in)
int n1, n2;
String choice ;  // just choice declaration
do
{   // do while body starts
```

Sentinel Condition do while loop…

```
System.out.println("Enter 1st no");
n1=input.nextInt();
System.out.println("Enter 2nd no");
n2=input.nextInt();
System.out.println("Sum="+(n1+n2));
System.out.println("Do you want to do more addition (Yes/No)");
choice= input.next();
} // do while loop body closed
while(choice.equals("Yes"));
System.out.println("Thank you for using do while loop");
} // main method body closed
} // class body closed
```

Sentinel Condition do while loop…

Note:

1) In the previous example all choices except “Yes” will be considered no and loop will be terminated.
2) In this do while loop example we do not initialize string at top because condition is checked at the end.

Sentinel Condition do while loop Task

1. Make a calculator using sentinel condition do while loop and if else if ladder statement which perform the following addition, subtraction, multiplication, division and remainder value. Take values of operator, numbers and choice from user on runtime.

```
else if(operator.equals("/"))
{
    if(num2==0)
    {
    System.out.println("Cannot divided by 0");
    }
    else
    System.out.println("Division is : " +(num1/num2));
}
```

Nested Loop

Loop within the body of another loop is called nested loop.

Nested for loop

Program 1: (Triangular loop which make triangle using for nested loop with help of astricts (*)).

```
*
**
***
****
*****
```

Nested for loop…

```
class NestedForDemo1
{
public static void main(String args[])
{
int i, j;
```

Nested for loop…

```
for (i=1 ; i<=5 ; i++ )
        {
        for (j=1 ; j<=i ; j++ )
                {
                  System.out.println(“*”);  //print *
                }
        System.out.println(“”);    // used for new line
}
}  }
```

Nested for Loop…

Program 2: (Triangular loop which make triangle using for nested loop with help of numbers).

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

Nested for loop…

```
class NestedForDemo2
{
public static void main(String args[])
{
int i, j;
```

Nested for loop

```
for (i=1 ; i<=5 ; i++ )
        {
        for (j=1 ; j<=i ; j++ )
                {
                  System.out.println(j);
                  System.out.println(“  ”); //for blank space
                }
        System.out.println(“”);    // used for new line
}
}  }
```

Nested for loop tasks

1. Write a java program that will display * in the following pattern.

```
*****
****
***
**
*
```

2. Write a java program that will display numbers in the following pattern.

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

Java Random Class

- Random class is part of java.util package.
- An instance of java Random class is used to generate random numbers.
- This class provides several methods to generate random numbers of type integer, double, long, float etc.

Java Random Class Methods

1) nextBoolean(): This method returns next pseudorandom which is a boolean value from random number generator sequence.

2) nextDouble(): This method returns next pseudorandom which is double value between 0.0 and 1.0.

3) nextFloat(): This method returns next pseudorandom which is float value between 0.0 and 1.0.

Java Random Class Methods...

4) nextInt(): This method returns next int value from random number generator sequence.

5) nextInt(int n): This method return a pseudorandom which is int value between 0 and specified value from random number generator sequence.

Java Random Example

```
import java.util.Random;

/**
 * Java Random Number Example Program
 *
 */
public class RandomNumberExample {

        public static void main(String[] args) {

                //initialize random number generator
                Random random = new Random();

                //generates boolean value
                System.out.println(random.nextBoolean());
```

Java Random Example…

```
                //generates double value
                System.out.println(random.nextDouble());

                //generates float value
                System.out.println(random.nextFloat());

                //generates int value
                System.out.println(random.nextInt());

                //generates int value within specific limit
                System.out.println(random.nextInt(20));

        }

}
```

Java Random Example...

Output of the above program is:

```
false
0.30986869120562854
0.6210066
-1348425743
18
```

Random Numbers

```
import java.util.Random;

Random rand = new Random();

int num = rand.nextInt(6)+1;
```

Here 6 means it will generate numbers from 0 to 5 means it will generate 6 numbers.

Game of Random Numbers

```
import java.util.Random;
import java.util.Scanner;
public class Game
{           // Class body starts
Public static void main(String args[])
{           // main method body starts
int num, guess, count=0;
Scanner input = new Scanner(System.in);
Random rand = new Random();
```

Game of Random Numbers

```
num = rand.nextInt(100)+1;
System.out.println(“Number Generated, try to guess it”);
while(true)
{
count++;
System.out.println(“Enter your guess: ”);
guess = input.nextInt();
if(guess>num)
{
System.out.println(“Your number is high, please try again”);
}
```

Game of Random Numbers

```
else if(guess<num)
{
System.out.println(“Your number is low, please try again”);
}
else
{
break;
}
}            // while loop body closed
```

Game of Random Numbers

```
System.out.println(“You found the guess in” + count+ “Attempts”);
if(count<5)
{
System.out.println(“Excellent”);
}
if(count>5)
{
System.out.println(“Good”);
}

} // Main method body closed
} // class body closed
```

THANK YOU