# Iterative statements

for loop

## Two types of repetition

- Number of repetitions is ***<u>known</u>***
  - *for* loop
- Number of repetitions is ***<u>unknown</u>***
  - *while* and *do-while* loop

## Number of repetitions is ***known***

Examples in real life:

```
Repeat 10 times
    Do a press up

For each customer from 1 to number in group
    Take food order
    Take drinks order
```

## Number of repetitions is ***known***

Examples in java programs:

```
For the days from 1 to 7
    Ask user to enter overtime hours
    Calculate overtime pay

For the years from 1 to mortgage term
    Calculate interest due on mortgage
```

## `for` loop

- The ***for*** loop is used to execute one or more statements a *specified* number of times
- There are 3 parts to a for loop

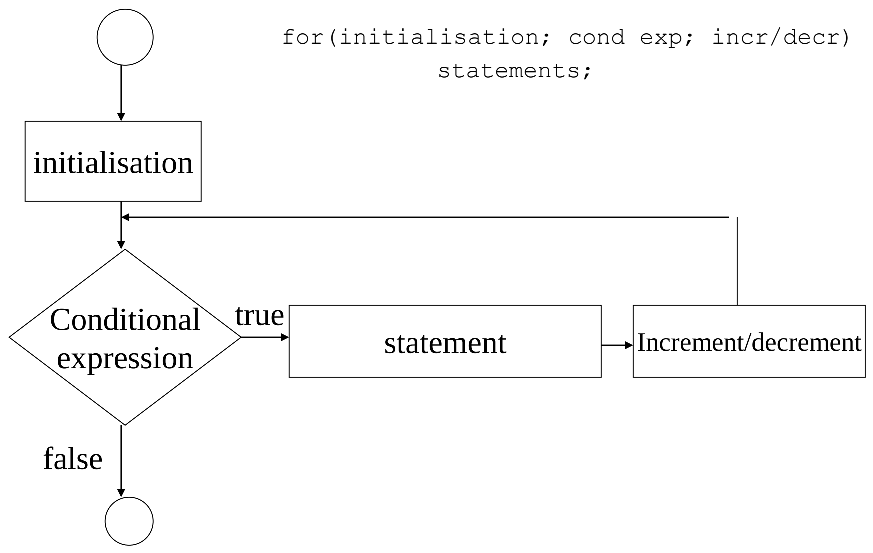
```
for(initialization; conditional test; increment/dec)
{
    statement;
}
```

## `for` Loop

- The ***initialization*** section is executed only once. This is where the loop will start counting.
- The loop will repeat if the ***conditional test*** is true and terminate when it becomes false
  - ***Conditional test*** performed at the start or top of the loop each time the loop is repeated
- The ***increment/decrement*** is executed at the end of each pass through the loop. This is how the loop counts.

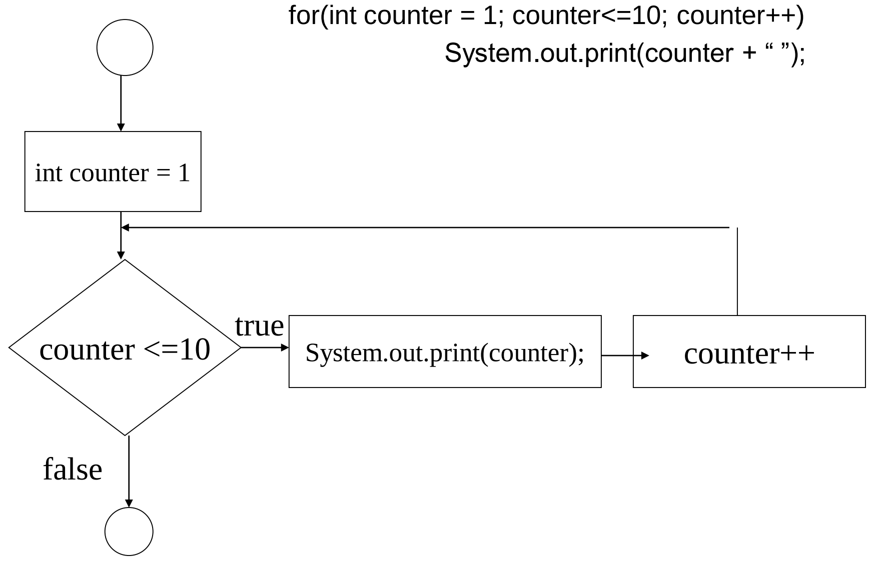
## for Flowchart

```
for(initialisation; cond exp; incr/decr)
           statements;
```



## for Flowchart

```
for(int counter = 1; counter<=10; counter++)
           System.out.print(counter + " ");
```



## Sample Program 1

```
//prints numbers 1 to 10
public class ForSample1
{
   public static void main(String[] args)
   {
      for(int count = 1; count <= 10; count++)
      {
         System.out.print(count + " ");
      } //end for
   } //end main method
} //end class
```

## 3 parts of `for` loop

|  |  |
|---|---|
| `int count = 1` | initialization |
| `count <= 10` | conditional test |
| `count ++` | increment |

## `for` loop

1. Initialize loop control variable
2. Evaluate conditional test
   – Should conditional test be true, execute statements in loop, and update loop control variable. Go back to step 2.
3. Should conditional test be false, loop terminates and next line of code is executed

## Sample Program 2

```
/prints numbers 10 to 1 and then Blastoff
public class ForSample2
{
   public static void main(String[] args)
   {
     for(int count = 10; count >= 1; count--)
     {
        System.out.println(count);
     }

     System.out.println("blastoff!!!");

   } //end main method
} //end class
```

## Sample Program 3

```
//find sum of numbers from 1 to 5
public class ForSample3
{
  public static void main(String[] args)
  {
     int total = 0;
     for(int count = 1; count <= 5; count++)
     {
        total = total + count;  //add current value of count to total
     }
     System.out.println("total of nos 1 to 5 is " +total);
   } //end main method
} //end class
```

## Accumulating Values

- Often you need to total or sum values
- Set the `total` variable to 0
- The accumulation instruction

  `total = total + count;`

  tells the computer to add the value of `count` to the old value of `total` and to store the result in `total`

- The value of `count` is incremented allowing progression from one value to the next until the end of the loop.

## Example - Find the average of *n* numbers

- GET n (no of values to average)
- For each number (repeat n times)
  - GET number
  - ADD number to total
- Calc average
- Display average

//program to find average of n values

```
import java.util.*;
public class AverageNValues
{
   public static void main(String[] args)
   {
      Scanner keyIn = new Scanner(System.in);
      int num, n, total = 0;
      double average;
      System.out.print("How many values do you want to enter? ");
      n = keyIn.nextInt();
                 //repeat n times
      for(int count = 1; count <= n; count++)
      {
         System.out.print("Enter value " +count +": " );
         num = keyIn.nextInt();
         //add num to total
         total = total + num;
      }
      //calc average
      average = total/n;
      //display average
      System.out.println("Average is  " +average);

  } //end main method
} //end class
```