**Looping / Iteration:**

1. **Pre-Tested Loop /Entry Control**
2. **For loop**
3. **While**
4. **Extended for / foreach loop**
5. **Post Tested Loop /Exit Control**
6. **Do while loop**

**Flow Control Statement :** These statements are used to control the execution of the program, we have the following types of flow controls statements :

**1. Conditional Statements :** Using these statements we can make the decision such as using if else, nested if, if else if ladder etc.

**2. Looping Constructs :** Using these statements we can repeat the given statement.

**Jump Statement :** These statements are used to skip or jump the control flow of execution inside the loop.

(a) **continue :** when the continue keyword is encountered inside the loop it will skip the rest of the lines after it and will send the control back in the begining of the loop.

(b) **break :** when the break keyword is encountered inside the loop it will terminate the loop pre maturely.

WAP to accept name, age and gender and decide for the voting rights and marriage

Display message like Mr./Miss Name you can vote but cannot marry.

**Example of for loop: Increment for loop**

class abc

{

public static void main(String args[])

{

for(int i=0;i<5;i++)

{

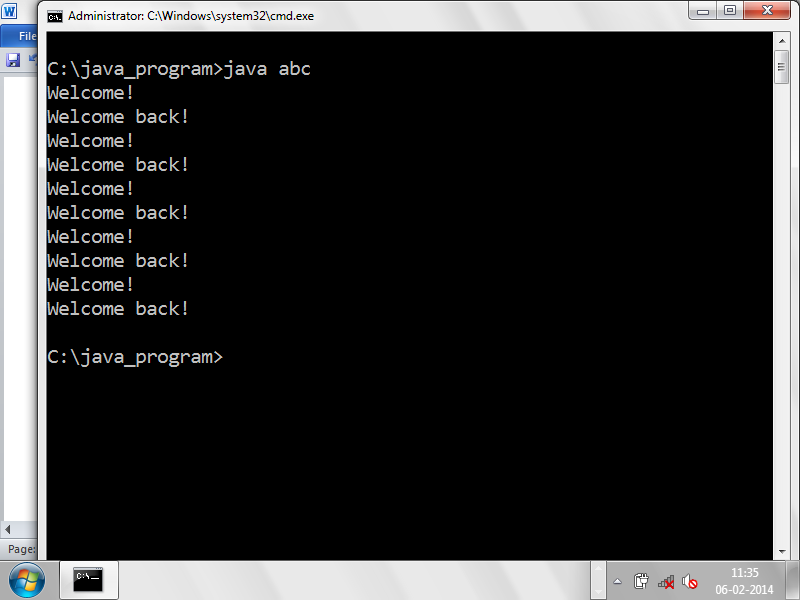
System.out.println("Welcome!");

System.out.println("Welcome back!");

}

}

}



**Decrement for loop:**

class abc

{

public static void main(String args[])

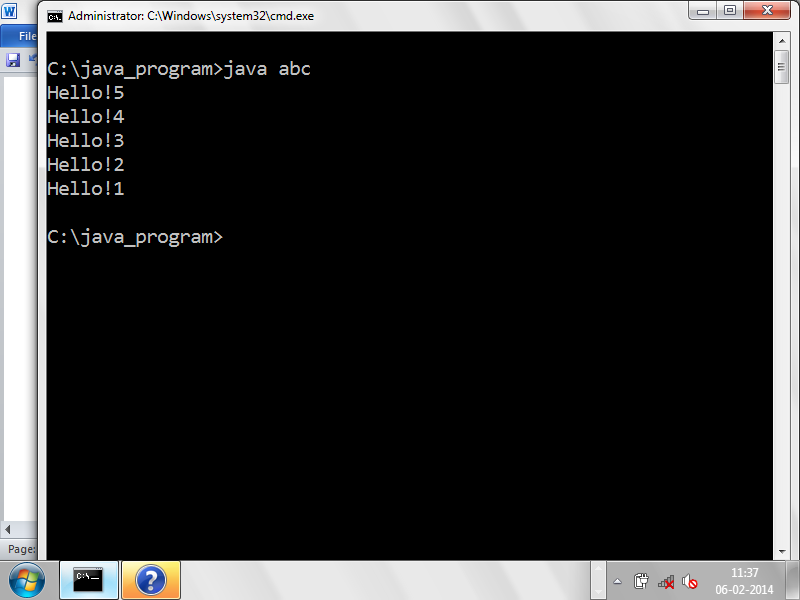
{

for(int i=5;i>0;i--)

System.out.println("Hello!" + i);

}

}



**Example of Nested for loop:**

class abc

{

public static void main(String args[])

{

for(int i=0;i<5;i++)//outer loop

{

System.out.println("Outer : " + i);

for(int j=0;j<5;j++)

{

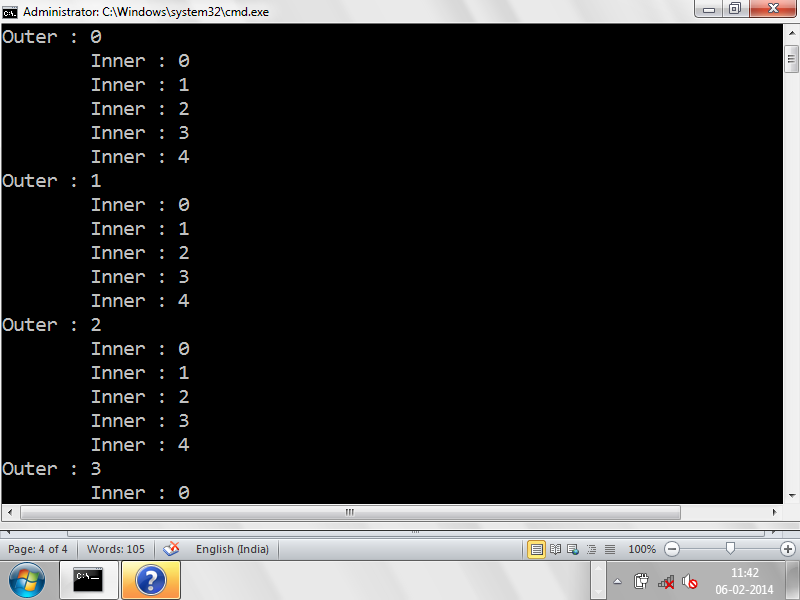
System.out.println("\tInner : " + j);

}

}

}

}



**Continue keyword:** it is used to skip or bypass the statement in the loop if the condition is true, it will send the control back to the loop rather than print the next line

**Example:**

class abc

{

public static void main(String args[])

{

for(int i=0;i<5;i++)

{

if(i==3 || i==2)

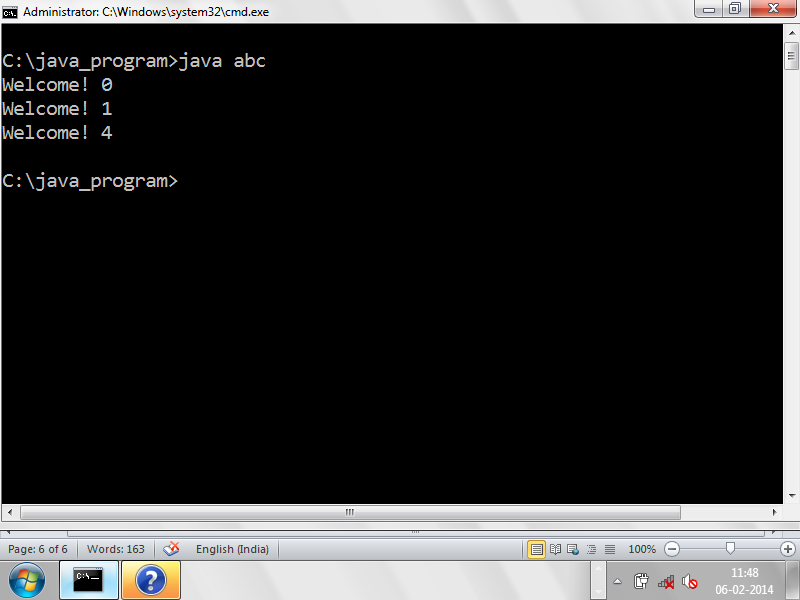
continue;

System.out.println("Welcome! " + i);

}

}

}



**Break keyword:** it is used to terminate the loop prematurely when the given condition is satisfied.

**Example:**

class abc

{

public static void main(String args[])

{

for(int i=0;i<5;i++)

{

if(i==3)

{

System.out.println("terminated at : " + i);

break;

}

System.out.println("Welcome! " + i);

}

}

}

**Example of continue in the nested for:**

class abc

{

public static void main(String args[])

{

for(int i=0;i<5;i++)

{

for(int j=0;j<5;j++)

{

if(j==2) continue;

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

}

}

}

}

**Example of while loop:**

class abc

{

public static void main(String args[])

{

int x=0;

while(x<5)

{

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

x++;

}

}

}

**Do while loop:**

class abc

{

public static void main(String args[])

{

int x=10;

do

{

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

x++;

}while(x<5);

}

}

**Extended for:**

It is useful for printing all the elements of the given collection i.e. array

**Syntax:**

For(var type : collection)

**Example:**

class abc

{

public static void main(String args[])

{

for(String s : args)

System.out.println(s);

}

}

**Example:**

class abc

{

public static void main(String args[])

{

int arr1[] = new int[5];

arr1[0] = 1111;

arr1[1] = 1112;

arr1[2] = 1113;

arr1[3] = 1114;

arr1[4] = 1115;

for(int s : arr1)

System.out.println(s);

}

}