```java
/*for loop: for loop is the best choice if we know the number of iterations in advance.
It is also called entry controlled loop.
Syntax:
for(initialization;condition;increment/decrement)
{
    Statement or code to be executed;
}
*/
/*-------------------------------------------------------------------------------*/

/*#1WAP to print numbers from 1 to 10 using for loop.*/
import java.util.*;
class Test1
{
    public static void main(String[] args)
    {
        for(int i=1;i<=10;i++)
        {
            System.out.print(i+" ");
        }
    }
}
/* OUTPUT
1 2 3 4 5 6 7 8 9 10
*/
/*-------------------------------------------------------------------------------*/

/* #2 Curly Braces are optional. we can take only one statement under for loop
without curly braces. */
//import java.util.*;
class Test2
{
    public static void main(String[] args)
    {
        for(int i=1;i<=10;i++)

            System.out.print(i+" ");

    }
}
/* OUTPUT
1 2 3 4 5 6 7 8 9 10
*/
/*-------------------------------------------------------------------------------*/

/* #3 Curly Braces are optional. we can take only one statement under for loop
without curly braces which should not be declarative statement. */
//import java.util.*;
class Test3
{
    public static void main(String[] args)
    {
        //for(int i=1;i<=10;i++)

            //int x;

    }
}
/* OUTPUT
L1.java:50: error: variable declaration not allowed here
                        int x=10;
                            ^
1 error
*/
/*-------------------------------------------------------------------------------*/

/* #4 We can take declarative statement under for loop using curly Braces. */
//import java.util.*;
class Test4
```

```
{
    public static void main(String[] args)
    {
        for(int i=1;i<=10;i++)
        {
            int x=10;
        }

    }
}
/* OUTPUT
No compiler error
*/
/*-------------------------------------------------------------------------------------*/

/* #5 Initialization section will be executed only once. Here we declare
and initialize local variables for for loop.
int i=0,j=0; can declare multiple variables but should be of same type.
int i=0,String s="Amit"; it is not valid because we can declare any number of variables
but
should be of same type.
int i=0,int j=0; it is also not valid. We need to write data type only once.*/
//import java.util.*;
class Test5
{
    public static void main(String[] args)
    {
        //for(int i=1, int j=10;i<j;i++,j--)
        {
            //System.out.print(i+j+" ");
        }

    }
}
/* OUTPUT
Compiler error
*/
/*-------------------------------------------------------------------------------------*/

/* #6 In the initialization section we can take any valid java statement including
sopln.*/
//import java.util.*;
class Test6
{
    public static void main(String[] args)
    {
        int i=0;
        for(System.out.print("HELLO");i<3;i++)
        {
            System.out.print(" HI");
        }

    }
}
/*OUTPUT
HELLO HI HI HI
*/

/*-------------------------------------------------------------------------------------*/

/* #7 In the Conditional section we can take any valid java statement but it should be
of boolean type. If we will not take anything in the conditional section then compiler
will place true in the conditional statement.*/
//import java.util.*;
class Test7
{
    public static void main(String[] args)
    {
        for(int i=1;;i++)
```

```
        {
            System.out.print("HELLO");
        }

    }
}
/*OUTPUT
HELLO will be printed infinite times.
*/
/*-------------------------------------------------------------------------------------*/

/* #8 In the Increment/Decrement section we can take any valid java statement including
sopln. We can take multiple increment/decrement variables also.*/
//import java.util.*;
class Test8
{
    public static void main(String[] args)
    {
        int i=0;
        for(System.out.print("HELLO");i<3;System.out.print(" HI"))
        {
            i++;
        }

    }
}
/*OUTPUT
HELLO HI HI HI.
*/
/*-------------------------------------------------------------------------------------*/

/* #9 All the three parts of for loop are independent of each other and are optional.*/
//import java.util.*;
class Test9
{
    public static void main(String[] args)
    {
        for(;;)
        {

        }

    }
}
/*OUTPUT
It is infinite loop and will not display anything.
*/
/*-------------------------------------------------------------------------------------*/

/* #10 All the three parts of for loop are independent of each other and are optional.*/
//import java.util.*;
class Test10
{
    public static void main(String[] args)
    {
        for(;;);

    }
}
/*OUTPUT
It is infinite loop and will not display anything.
*/
/*-------------------------------------------------------------------------------------*/

/* #11 Unreachability: in the following java parogram the body of the for loop will be
executed infinite times as in the conditional section true is written so the statements
after the body of the for loop will never get the chance for its execution. This concept
is called Unreachability. And in this case compiler will give  compiler Error.*/
//import java.util.*;
```

```
class Test11
{
    public static void main(String[] args)
    {
        for(int i=0;true;i++)
        {
            System.out.print("HELLO");
        }
    //  System.out.print("HI");

    }
}
/*OUTPUT
error: unreachable statement
                System.out.print("HI");
                ^.
*/
/*-------------------------------------------------------------------------------*/

/* #12 Unreachability: in the following java parogram the body of the for loop will not be
executed as in the conditional section false is written so the statements within the body
of the for loop will never get the chance for its execution. This concept is called
Unreachability. And in this case compiler will give  compiler Error.*/
//import java.util.*;
class Test12
{
    public static void main(String[] args)
    {
        /* Uncomment this block and see the result
        for(int i=0;false;i++)
        {
            System.out.print("HELLO");
        }
        */
        System.out.print("HI");

    }
}
/*OUTPUT
error: unreachable statement
                {
                ^.
*/
/*-------------------------------------------------------------------------------*/

/* #13 In the above two java parogram in conditional section boolean vaues true/false are
written which are known to compiler that's why it will give the concept of unreachability
but in case of any other valid condition in conditional section compiler will not evaluate
its boolean value at compile time hence in this case it will not give the concept of
unreachability.*/
//import java.util.*;
class Test13
{
    public static void main(String[] args)
    {
        int a=10,b=20;
        for(int i=0;a<b;i++)
        {
            System.out.print("HELLO");
        }

        System.out.print("HI");

    }
}
/*OUTPUT
```

```
Hello will be printed infinite times.
*/
/*-------------------------------------------------------------------------------*/

/* #14 In the above two java parogram (Concept 11 and 12) in conditional section boolean vaues true/false are written which are known to compiler that's why it will give the concept
of unreachability but in case of any other valid condition in conditional section compiler
will not evaluate its boolean value at compile time hence in this case it will not give the
concept of unreachability.*/
//import java.util.*;
class Test14
{
    public static void main(String[] args)
    {
        int a=10,b=20;
        for(int i=0;a>b;i++)
        {
            System.out.print("HELLO");
        }

        System.out.print("HI");

    }
}
/*OUTPUT
HI.
*/
/*while loop: while loop is the best choice if we do not know the number of iterations in advance.
It is also called entry controlled loop.
Syntax:
initialization
while(Condition of boolean type)
{
    Statement or code to be executed;
    Increment/Decrement;
}
*/
/*-------------------------------------------------------------------------------*/

/*#15 WAP to print numbers from 1 to 10 using while loop.*/
//import java.util.*;
class Test15
{
    public static void main(String[] args)
    {
        int i=1;
        while(i<=10)
        {
            System.out.print(i+" ");
            i++;
        }
    }
}
/* OUTPUT
1 2 3 4 5 6 7 8 9 10
*/
/*-------------------------------------------------------------------------------*/

/* #16 Curly Braces are optional. we can take only one statement under while loop without curly braces. */
//import java.util.*;
class Test16
{
    public static void main(String[] args)
    {
```

```
        while(true)
            System.out.print("Hello ");


    }
}
/* OUTPUT
Hello will be printed infinite times.
*/
/*-------------------------------------------------------------------------------------*/

/* #17 Curly Braces are optional. we can take only one statement under while loop
without curly braces which should not be declarative statement. */
//import java.util.*;
class Test17
{
    public static void main(String[] args)
    {
        //while(true)
        //  int x;

    }
}
/* OUTPUT
L1.java:357: error: variable declaration not allowed here
                        int x;
                            ^
1 error
*/
/*-------------------------------------------------------------------------------------*/

/* #18 We can take declarative statement under while loop using curly Braces. */
//import java.util.*;
class Test18
{
    public static void main(String[] args)
    {
        while(true)
        {
            int x=10;
        }

    }
}
/* OUTPUT
No compiler error
*/
/*-------------------------------------------------------------------------------------*/

/* #19 Unreachability: in the following java parogram the body of the while loop will be
executed infinite times as in the conditional section true is written so the statements
after the body of the while loop will never get the chance for its execution. This
concept
is called Unreachability. And in this case compiler will give  compiler Error.*/
//import java.util.*;
class Test19
{
    public static void main(String[] args)
    {
        while(true)
        {
            System.out.print("HELLO");
        }
        //System.out.print("HI");

    }
}
/*OUTPUT
L1.java:400: error: unreachable statement
```

```
                System.out.print("HI");
                ^
*/
/*-------------------------------------------------------------------------------------*/

/* #20 Unreachability: in the following java parogram the body of the while loop will not be
executed as in the conditional section false is written so the statements within the body
of the while loop will never get the chance for its execution. This concept is called
Unreachability. And in this case compiler will give  compiler Error.*/
//import java.util.*;
class Test20
{
    public static void main(String[] args)
    {
        /* Uncomment this block and see the result
        while(false)
        {
            System.out.print("HELLO");
        }
        */
        System.out.print("HI");

    }
}
/*OUTPUT
L1.java:422: error: unreachable statement
                {
                ^.
*/
/*-------------------------------------------------------------------------------------*/

/* #21 In the above two java parogram in conditional section boolean vaues true/false are
written which are known to compiler that's why it will give the concept of unreachability
but in case of any other valid condition in conditional section compiler will not evaluate
its boolean value at compile time hence in this case it will not give the concept of
unreachability.*/
//import java.util.*;
class Test21
{
    public static void main(String[] args)
    {
        int a=10,b=20;
        while(a<b)
        {
            System.out.print("HELLO");
        }

        System.out.print("HI");

    }
}
/*OUTPUT
Hello will be printed infinite times.
*/
/*-------------------------------------------------------------------------------------*/

/* #22 In the above two java parogram (Concept 19 and 20) in conditional section boolean
vaues true/false are written which are known to compiler that's why it will give the concept
of unreachability but in case of any other valid condition in conditional section compiler
will not evaluate its boolean value at compile time hence in this case it will not give the
concept of unreachability.*/
//import java.util.*;
class Test22
```

```
{
    public static void main(String[] args)
    {
        int a=10,b=20;
        while(a>b)
        {
            System.out.print("HELLO");
        }

        System.out.print("HI");

    }
}
/*OUTPUT
HI
*/
/*do while loop: If you want to execute loop body atleast once then you should go for
do while loop.
It is also called exit controlled loop.
Syntax:
Initialization
do
{
    Statement or code to be executed;
    Increment/Decrement;
}while(boolean type condition);
*/
/*-------------------------------------------------------------------------------------*/

/*#23 WAP to print numbers from 1 to 10 using do while loop.*/
//import java.util.*;
class Test23
{
    public static void main(String[] args)
    {
        int i=1;
        do
        {
            System.out.print(i+" ");
            i++;
        }while(i<=10);
    }
}
/* OUTPUT
1 2 3 4 5 6 7 8 9 10
*/
/*-------------------------------------------------------------------------------------*/

/* #24 Curly Braces are optional. we can take only one statement under do while loop
without curly braces. */
//import java.util.*;
class Test24
{
    public static void main(String[] args)
    {
        do
            System.out.print("Hello ");

        while(true);
    }
}
/* OUTPUT
Hello will be printed infinite times.
*/
/*-------------------------------------------------------------------------------------*/

/* #25 Curly Braces are optional. we can take only one statement under do while loop
without curly braces which should not be declarative statement. */
//import java.util.*;
```

```
class Test25
{
    public static void main(String[] args)
    {
        //do
        //  int x;
        //while(true);

    }
}
/* OUTPUT
L1.java:543: error: variable declaration not allowed here
                        int x;
                            ^
1 error
*/
/*-------------------------------------------------------------------------------*/

/* #26 We can take declarative statement under do while loop using curly Braces. */
//import java.util.*;
class Test26
{
    public static void main(String[] args)
    {
        do
        {
            int x=10;
        }
        while(true);

    }
}
/* OUTPUT
No compiler error
*/
/*-------------------------------------------------------------------------------*/

/* #27 Unreachability: in the following java parogram the body of the do while loop will be
executed infinite times as in the conditional section true is written so the statements
after the body of the do while loop will never get the chance for its execution. This concept
is called Unreachability. And in this case compiler will give  compiler Error.*/
//import java.util.*;
class Test27
{
    public static void main(String[] args)
    {
        do
        {
            System.out.print("HELLO");
        }while(true);
        //System.out.print("HI");

    }
}
/*OUTPUT
L1.java:588: error: unreachable statement
                System.out.print("HI");
                ^
1 error
*/
/*-------------------------------------------------------------------------------*/

/* #28 What will be the output of following java program?*/
//import java.util.*;
class Test28
{
    public static void main(String[] args)
```

```
    {

        do
        {
            System.out.print("HELLO ");
        }while(false);

        System.out.print("HI");

    }
}
/*OUTPUT
HELLO HI
*/
/*-------------------------------------------------------------------------------*/

/* #29 In the above 27th java parogram in conditional section boolean vaue true is
written which is known to compiler that's why it will give the concept of unreachability
but in case of any other valid condition in conditional section compiler will not evaluate
its boolean value at compile time hence in this case it will not give the concept of
unreachability.*/
//import java.util.*;
class Test29
{
    public static void main(String[] args)
    {
        int a=10,b=20;
        do
        {
            System.out.print("HELLO ");
        }while(a<b);

        System.out.print("HI");

    }
}
/*OUTPUT
Hello will be printed infinite times.
*/
/*-------------------------------------------------------------------------------*/

/* #30 In the above java parogram (Concept 27) in conditional section boolean
vaue true is written which is known to compiler that's why it will give the concept
of unreachability but in case of any other valid condition in conditional section compiler
will not evaluate its boolean value at compile time hence in this case it will not give the
concept of unreachability.*/
//import java.util.*;
class Test30
{
    public static void main(String[] args)
    {
        int a=10,b=20;
        do
        {
            System.out.print("HELLO ");
        }while(a>b);

        System.out.print("HI");

    }
}
/*OUTPUT
 HELLO HI
*/
/*-------------------------------------------------------------------------------*/
```

```java
/*
Transfer Statements:
1. break: #31 The break statement is used to terminate the loop immediately.
We can use break statement inside switch and inside the loops.
*/
class Test31
{
    public static void main(String[] args)
    {
        for(int i=1;i<=10;i++)
        {
            if(i==5)
            {
                break;
            }
            else
            {
            System.out.print(i+ " ");
            }
        }
    }
}
/*OUTPUT
 1 2 3 4
*/
/*-------------------------------------------------------------------------------------*/

/*
Transfer Statements:
2. continue: #32 The continue statement is used to skip the current iteration of the
loop.
We can use continue statement inside the loops.
*/
class Test32
{
    public static void main(String[] args)
    {
        for(int i=1;i<=10;i++)
        {
            if(i==5)
            {
                continue;
            }
            else
            {
            System.out.print(i+ " ");
            }
        }
    }
}
/*OUTPUT
 1 2 3 4 6 7 8 9 10
*/
```