جامعة التنمية البشرية
UNIVERSITY OF HUMAN DEVELOPMENT

COMPUTER PROGRAMMING CONCEPTS

CS&IT 1101
Instructor: Shakar H. Salih
E-mail: shakar.salih@uhd.edu.iq

LECTURE 11: REPETITION IN JAVA

OUTLINES

- While loop

INTRODUCTION (1/3)

- Java's Iteration statements are , these statements are commonly called as loops. A loop repeatedly executes the same set of instructions until a termination condition is met.

Problems Without Loops

- For example, if we want to print Hello for 10 times, we need to call the print statement 10 times.

INTRODUCTION (2/3)

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

We print same output 10 time. It is very time consuming

PROBLEM WITHOUT LOOP DISADVANTAGE

1. It is very time consuming, inefficient, error prone and has lots of duplicate code.
2. If we want print Hello World instead of Hello, we have to change all the ten lines.
3. If we want print **100** lines more or **1000** lines more, we have to add so many more lines. It is very time consuming to do it.
4. What if the number of lines to be printed is given by the user and we do not know the value until the program runs. If user wants 7 lines, we should print 7, if he wants 12 we should print 12.

INTRODUCTION (3/3)

- Consider the following declarations:

```
int a, b, c, d, e, f, g, h, i, j;
```

Issues:

1. Variables of the same **data_type**, i.e. **int**.
2. Individual operation (e.g., input) is needed for each variable

Better Solution:

Use Loops. (**One** variable will be enough for the **10** input values).

LOOPS IN JAVA

- loop cause a **certain piece** of program to be executed a certain number of times. Consider this scenario:
- You want to execute some code/s certain number of times depending upon input from user.

This type of task can be solved in programming using loops.

REPETITION (LOOPS)

- Loops sometime also called **Repetition**.
- **Repetition** structures, or loops, are used when a program needs to **repeatedly process one** or **more** instructions until some condition is met, at which time the loop ends.
- **Repetition** allows the programmer to efficiently use variables.
- Every loop can be defined by three way:
  1. Starting Point
  2. Ending Point
  3. Sequence of Moving

WHILE LOOP IN JAVA

- Executes from **zero** to many times, **depending** on expression.
- The **while loop** checks whether the **test** expression is **true** or **not**. If it is true, code/s inside the body of while loop is executed, that is, code/s inside the braces { } are executed. Then again the test expression is checked whether test expression is true or not. **This process continues until the test expression becomes false.**

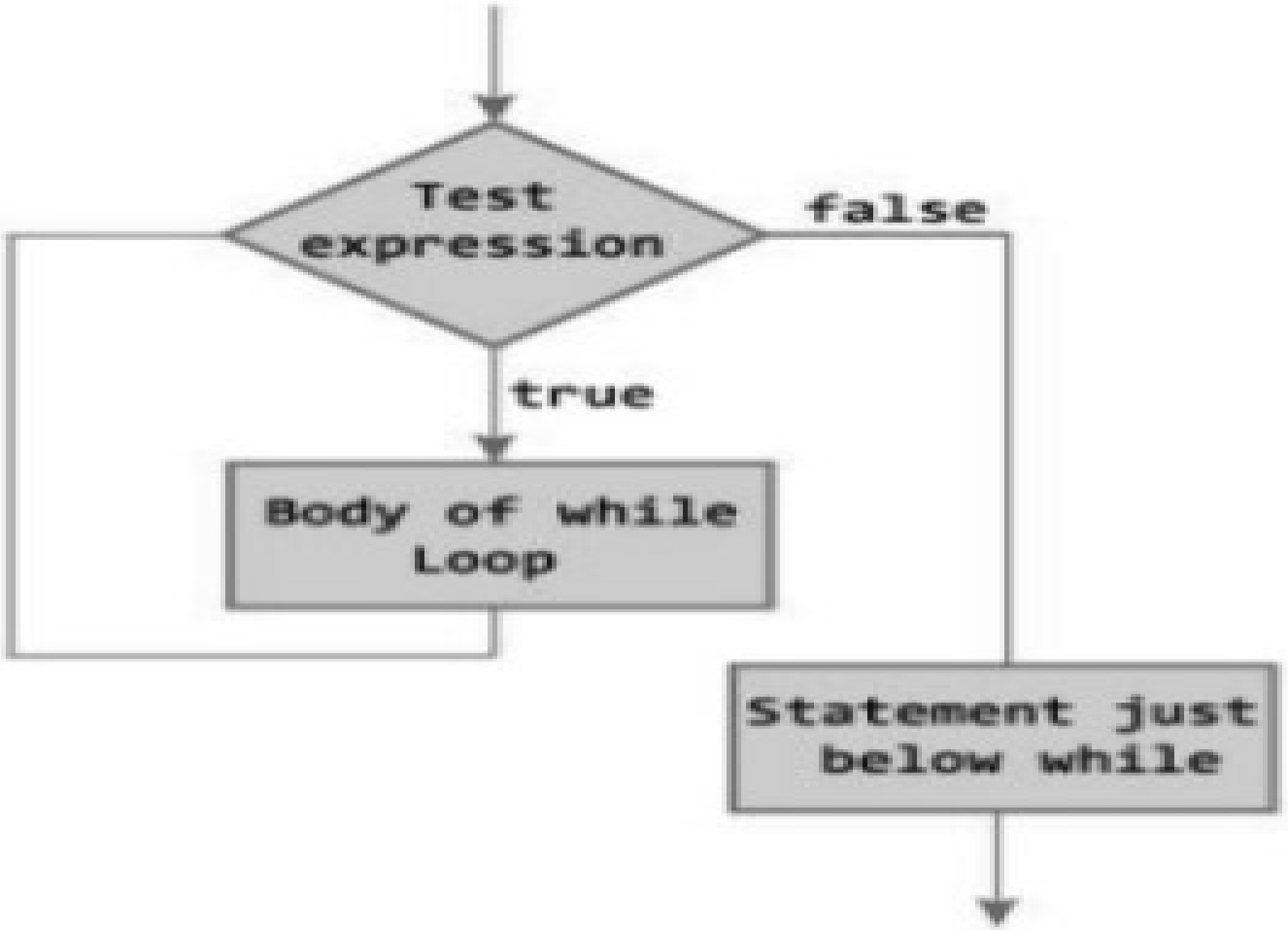
WHILE LOOP DECLARATION

Syntax:

```
while(test condition)
{
    block of code;
}
```

The **test condition** must be enclosed in parentheses. The **block of code** is called the body of the loop and is enclosed in braces and indented for readability. (The braces are not required if the body is composed of only ONE statement.) Semi-colons follow the statements within the block only

FLOW CHART OF WHILE LOOP



EXAMPLE 1 (1/6)

```
public class Test {
   public static void main(String[] args) {
      int x = 10;

      while( x < 15 ) {
         System.out.print("value of x : " + x );
         x++;
         System.out.print("\n");
      }
   }
}
```

Output is:

```
value of x : 10
value of x : 11
value of x : 12
value of x : 13
value of x : 14
```

Start point of while loop

Ending point of while loop

Print Value of x

Sequence of while loop

\n use of new line

EXAMPLE 1 (2/6)

```
int x = 10;
   while( x < 15 ) {
      System.out.print("value of x : " + x );
      x++;
      System.out.print("\n");
```

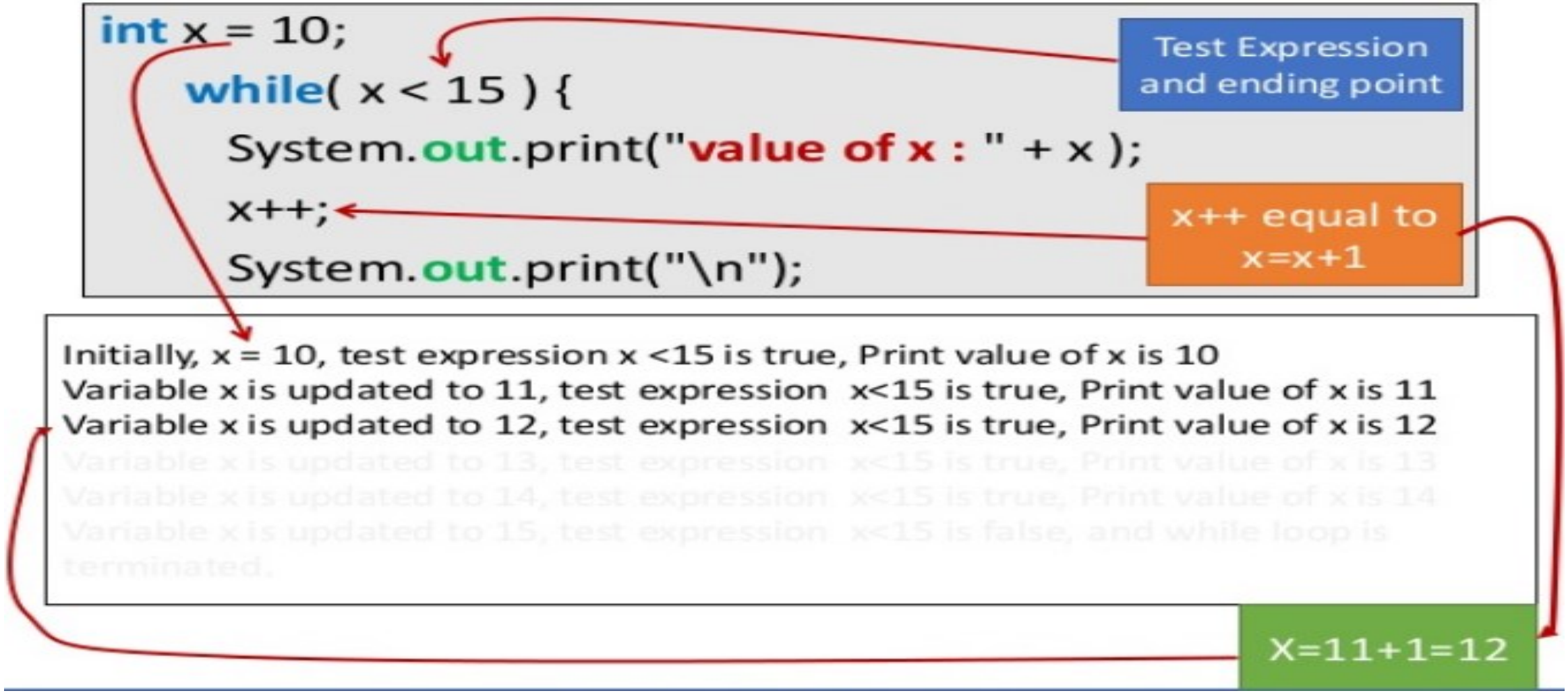
Test Expression and ending point

x++ equal to x=x+1

Initially, x = 10, test expression x <15 is true, Print value of x is 10
Variable x is updated to 11, test expression x<15 is true, Print value of x is 11
Variable x is updated to 12, test expression x<15 is true, Print value of x is 12
Variable x is updated to 13, test expression x<15 is true, Print value of x is 13
Variable x is updated to 14, test expression x<15 is true, Print value of x is 14
Variable x is updated to 15, test expression x<15 is false, and while loop is terminated.

X=10+1=11

EXAMPLE 1 (3/6)


int x = 10;
while(x < 15) {
System.out.print("value of x : " + x);
x++;
System.out.print("\n");
Test Expression and ending point
x++ equal to x=x+1
Initially, x = 10, test expression x <15 is true, Print value of x is 10
Variable x is updated to 11, test expression x<15 is true, Print value of x is 11
Variable x is updated to 12, test expression x<15 is true, Print value of x is 12
X=11+1=12

EXAMPLE 1 (4/6)

```
int x = 10;
   while( x < 15 ) {
      System.out.print("value of x : " + x );
      x++;
      System.out.print("\n");
```

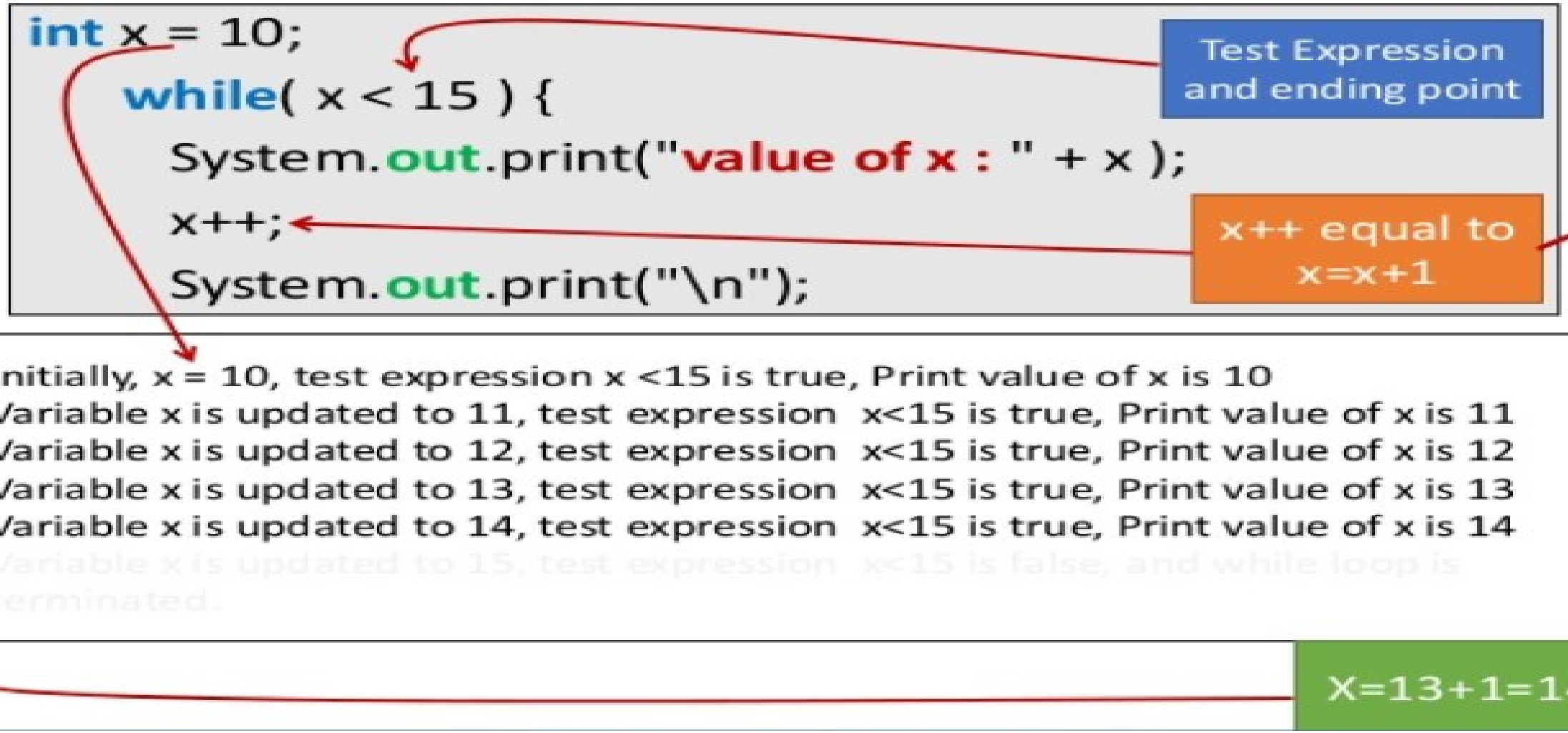
Test Expression and ending point

x++ equal to x=x+1

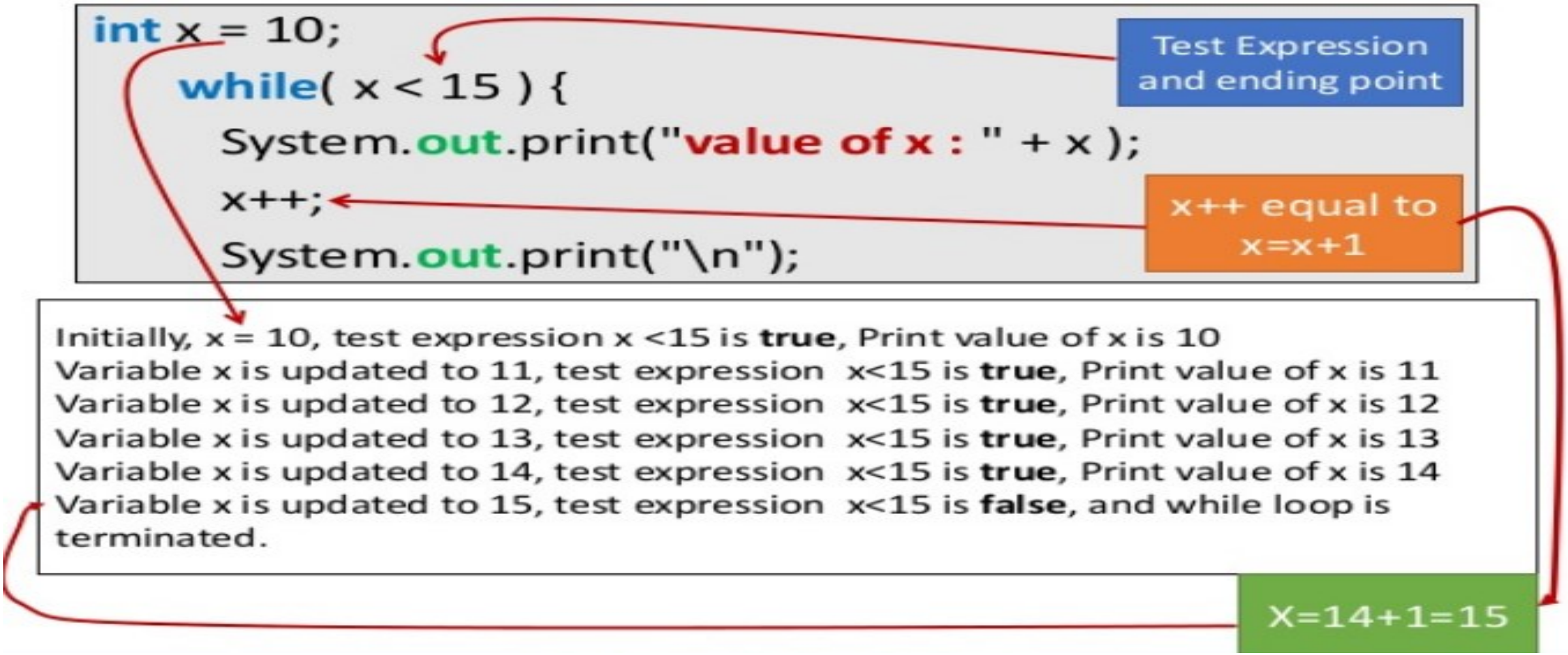
Initially, x = 10, test expression x <15 is true, Print value of x is 10
Variable x is updated to 11, test expression x<15 is true, Print value of x is 11
Variable x is updated to 12, test expression x<15 is true, Print value of x is 12
Variable x is updated to 13, test expression x<15 is true, Print value of x is 13
Variable x is updated to 14, test expression x<15 is true, Print value of x is 14
Variable x is updated to 15, test expression x<15 is false, and while loop is terminated.

X=12+1=13

EXAMPLE 1 (5/6)


int x = 10;
while(x < 15) {
System.out.print("value of x : " + x);
x++;
System.out.print("\n");
Test Expression and ending point
x++ equal to x=x+1
Initially, x = 10, test expression x <15 is true, Print value of x is 10
Variable x is updated to 11, test expression x<15 is true, Print value of x is 11
Variable x is updated to 12, test expression x<15 is true, Print value of x is 12
Variable x is updated to 13, test expression x<15 is true, Print value of x is 13
Variable x is updated to 14, test expression x<15 is true, Print value of x is 14
X=13+1=14

EXAMPLE 1 (6/6)


int x = 10;
while(x < 15) {
System.out.print("value of x : " + x);
x++;
System.out.print("\n");
Test Expression and ending point
x++ equal to x=x+1
Initially, x = 10, test expression x <15 is true, Print value of x is 10
Variable x is updated to 11, test expression x<15 is true, Print value of x is 11
Variable x is updated to 12, test expression x<15 is true, Print value of x is 12
Variable x is updated to 13, test expression x<15 is true, Print value of x is 13
Variable x is updated to 14, test expression x<15 is true, Print value of x is 14
Variable x is updated to 15, test expression x<15 is false, and while loop is terminated.
X=14+1=15

EXAMPLE 2

```
public class Test {
    public static void main(String[] args) {
        int i=5;
        while(i>=1){
            System.out.println(i);
            i--;
        }
    }
}
```

Output is:

```
5
4
3
2
1
```

Starting Point of while loop

Test Expression and Ending point of while loop

Print the value of i

Sequence of while loop, Decrement by 1

PROBLEM STATEMENT

- Java program to find factorial of a positive integer entered by user. (Factorial of n = 1*2*3...*n).
- Write a java program which displays odd numbers between 1 and 10.

INFINITE WHILE LOOP

```
public class Test {
    public static void main(String args[]){
        int i=10
        while(i>1)
        {
            System.out.println(i);
             i++;
        }
    }
}
```

This loop would never end, its an infinite while loop. This is because condition is i>1 which would always be true as we are incrementing the value of i inside while loop.

Avoid to compiler infinite Loops

THANK YOU.....

DO YOU HAVE ANY QUESTIONS ?