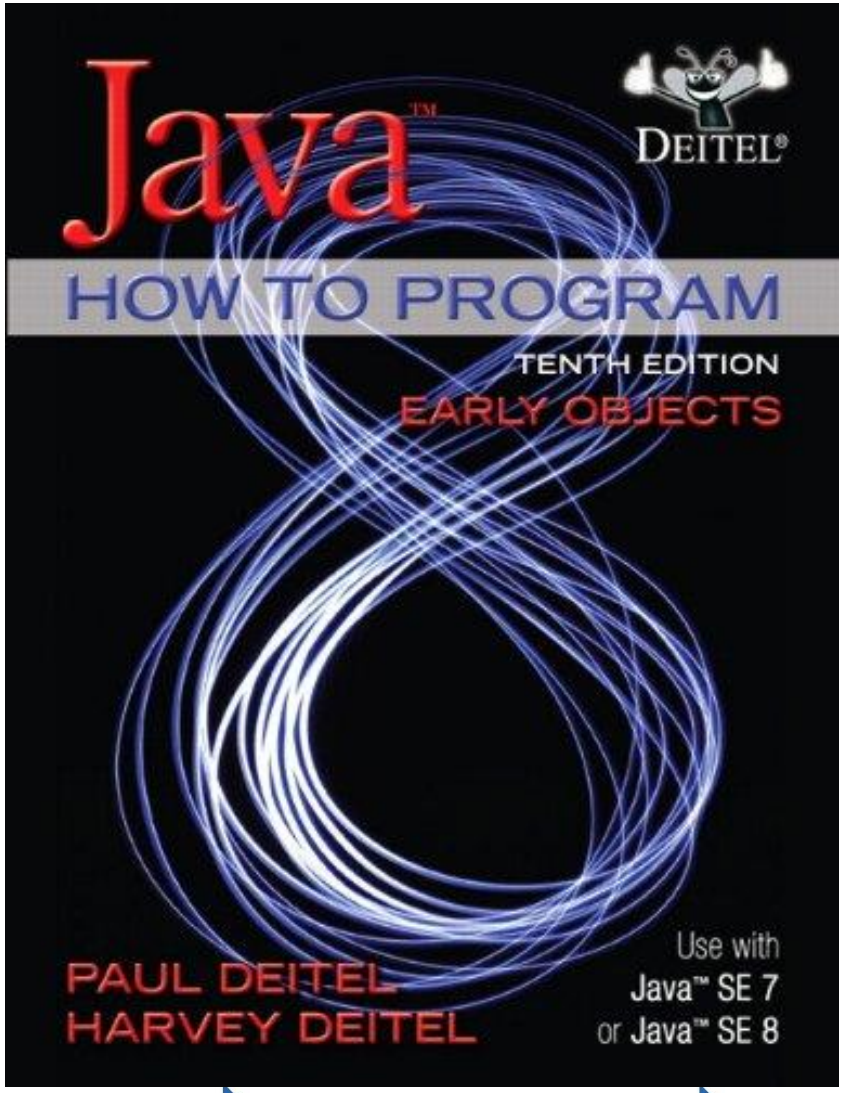
پوهنتون کاردان
KARDAN UNIVERSITY

Object Oriented Programming (Java)

Loops in Java:
While loop
&
Do-While loop

Text Book



Title: Java How to Program, Early Objects

Author(s): Paul Deitel, Harvey Deitel

Publisher: Pearson Education

Year: 2015

ISBN: 0133807800,9780133807806

Object Oriented Programming using Java by Simon Kendal

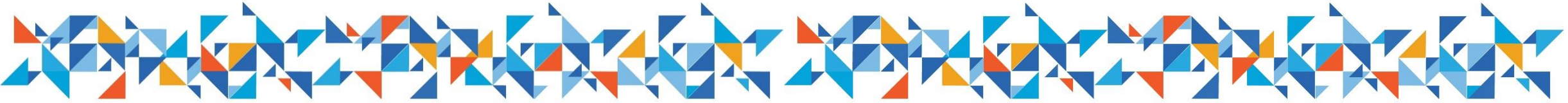
Learning Outcomes

پوهنتون کاردان
KARDAN UNIVERSITY

- **Students will be able to understand**

- **While Loop**

- **Do-while Loop**

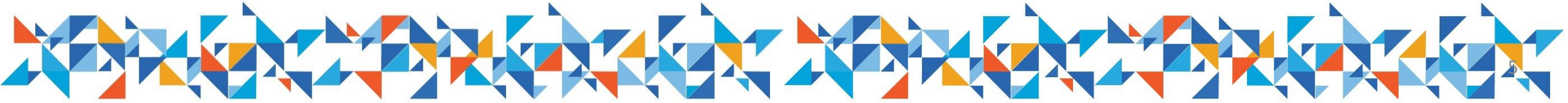


While loop

- The Java *while loop* is used to iterate a part of the program repeatedly until the specified Boolean condition is true.
- As soon as the Boolean condition becomes false, the loop automatically stops.
- The while loop is considered as a repeating if statement.
- If the number of iteration is not fixed, it is recommended to use the while loop.

Syntax:

```
while (condition){
//code to be executed
I ncrement / decrement statement
}
```

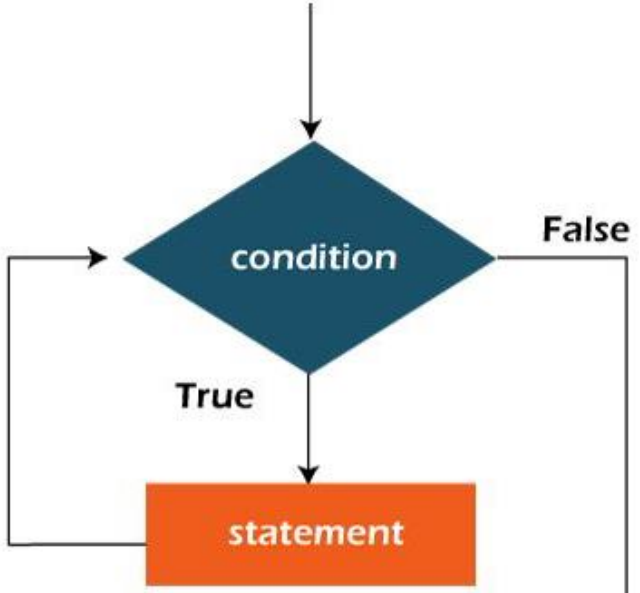


The different parts of while loop:

- 1. Condition: It is an expression which is tested. If the condition is true, the loop body is executed and control goes to update expression. When the condition becomes false, we exit the while loop.
- **Example**:
- **i <=100**
- 2. Update expression: Every time the loop body is executed, this expression increments or decrements loop variable.
- **Example:**
- **i++;**

Flowchart of Java While Loop

- Here, the important thing about while loop is that, sometimes it may not even execute.
- If the condition to be tested results into false, the loop body is skipped and first statement after the while loop will be executed.



Example 1:

- In the below example, we print integer values from 1 to 10. Unlike the for loop, we separately need to initialize and increment the variable used in the condition (here, i). Otherwise, the loop will execute infinitely.

```
//A program to print numbers from 1 to 10
public class whileExample {
    public static void main(String[] args) {

    int i=1;

    while(i<=10){
            System.out.println(i);
            i++;
}}}
```

Example 2:

```
//A program to print numbers from 10 to 1
public class infinite {
    public static void main(String[] args) {

    System.out.println(" printing numbers in reverse order");

    int i=10;
    while(i>=0){
        System.out.println(i);
        i--;
}}}
```

Example 3:

```
// This program will print " I am Kardanian" 20 times.
public class whileExample {
    public static void main(String[] args) {

    int i=1;

    while(i<=20){
        System.out.println("I am kardanian");
        i++;
}
    }}
```

Example 4:

```
//A program to print the summation of numbers from 1 to 10
public class whileExample {
    public static void main(String[] args) {

    int i=1, sum=0;

    while(i<=10){
            sum = sum + i;

            i++;
}
System.out.println("The summation of numbers from 1 to 10 is: "+ sum);
}}
```

Example 5:

```java
//This program finds the average of 5 subjects entered by the user
//Using while loop
import java.util.Scanner;
public class average_of_subjects {
    public static void main(String[] args) {
        Scanner obj=new Scanner (System.in);
        int total=0;
        int total_subjects=1;

        while(total_subjects<=5){
            System.out.println("Enter marks for subject # "+total_subjects);
            int marks=obj.nextInt();
            total=total+marks;
            total_subjects++;
        }
        int average= total/5;
        System.out.println("The average for the above 5 subjects is: "+average);
    }
}
```

Java Infinitive While Loop

- If you pass **true** in the while loop, it will be infinitive while loop.

Syntax:

```
while(true){
//code to be executed
}
```

Example for infinite loop

```
//This is an example for infinite loop
public class infinite {
    public static void main(String[] args) {

    while(true){
        System.out.println("this loop will not stop");
}
}}
```

do-while loop

- The Java *do-while loop* is used to iterate a part of the program repeatedly, until the specified condition is true.
- If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use a do-while loop.
- Java do-while loop is called an **exit control loop**.
- Therefore, unlike while loop and for loop, the do-while check the condition at the end of loop body.
- The Java *do-while loop* is executed at least once because condition is checked after loop body.

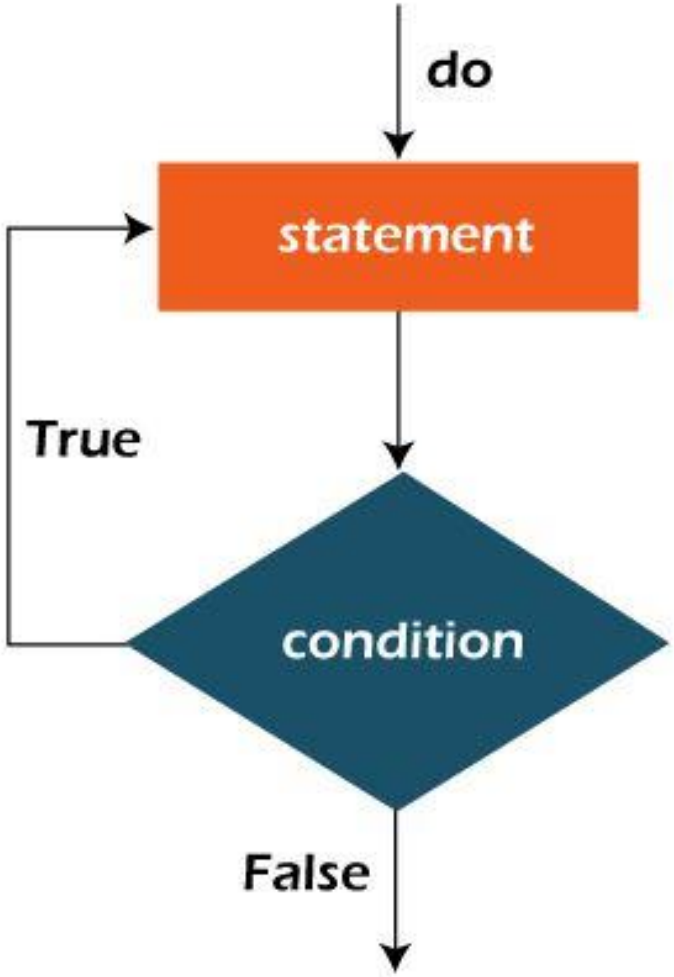
Syntax:

```
do{
//code to be executed / loop body
//update statement
}while (condition);
```

The different parts of do-while loop:

- 1. Condition: It is an expression which is tested. If the condition is true, the loop body is executed and control goes to update expression. As soon as the condition becomes false, loop breaks automatically.
- **Example:**
- **i <=100**
- 2. Update expression: Every time the loop body is executed, this expression increments or decrements loop variable.
- **Example:**
- **i++;**
- **Note:** The do block is executed at least once, even if the condition is false.

Flowchart of do-while loop



Example 1:

- In the below example, we print integer values from 1 to 10. Unlike the for loop, we separately need to initialize and increment the variable used in the condition (here, i). Otherwise, the loop will execute infinitely.

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

Java Infinitive do-while Loop

- If you pass **true** in the do-while loop, it will be infinitive do-while loop.

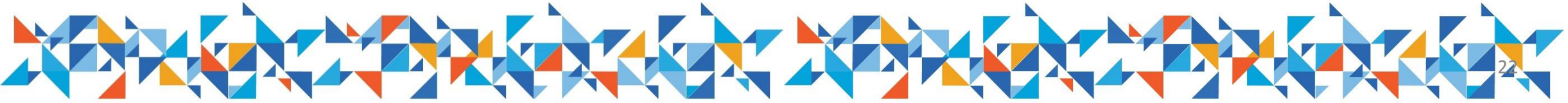
- **Syntax:**

```
do{
//code to be executed
}while(true);
```

```java
//This program will execute infinite times
//Using do-while loop
public class dowhileExample {
    public static void main(String[] args) {
        int i=1;
        do{
            System.out.println(i+"Infinitive do while loop");
            i++;
    }while(true);
}
}
```

پوهنتون کاردان
KARDAN UNIVERSITY

Write a program to print the table of 8, using do-while loop



```java
public class table_of_8_using_do_whileloop {
    public static void main(String[] args) {
        int table=8;
        int i=1;
        System.out.println("This is table of 8: ");
        do{
            System.out.println(i+" * 8 = "+ i*table);
            i++;
        }while(i<=10);
    }
}
```

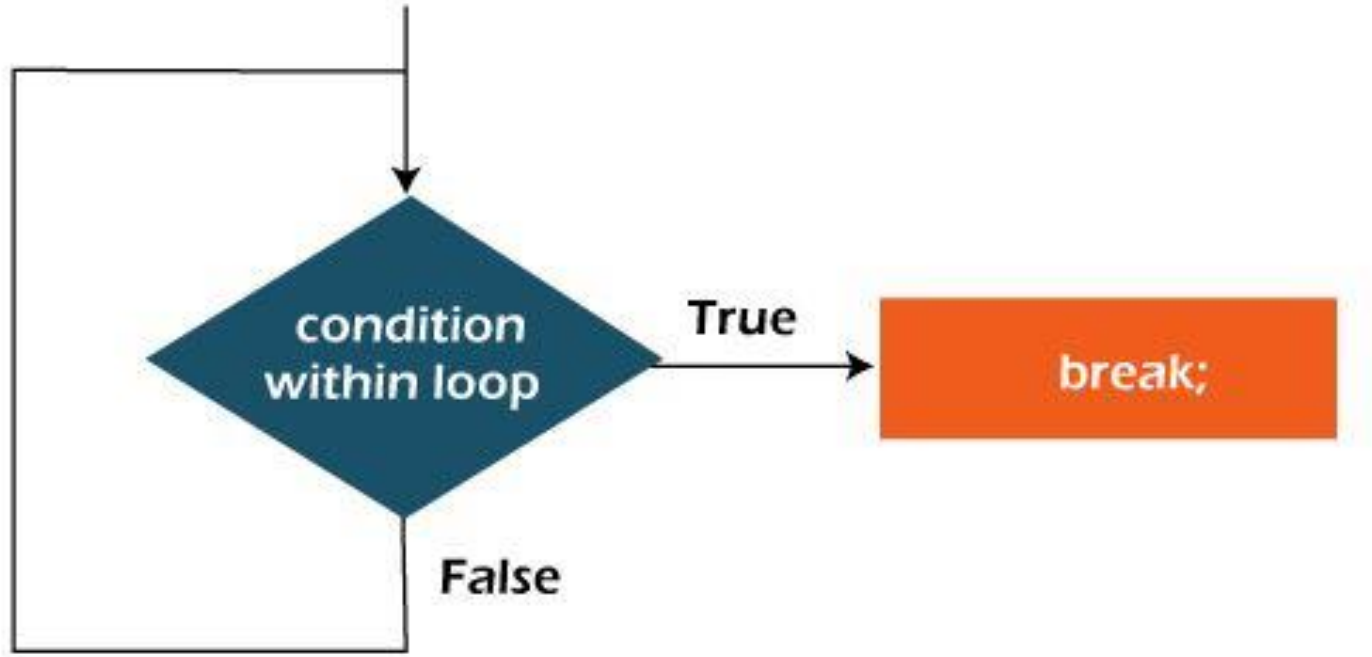
Java Break Statement

- When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.

- The Java *break* statement is used to break loop or switch statement. It breaks the current flow of the program at specified condition.

- We can use Java break statement in all types of loops such as for loop, while loop and do-while loop.

Flowchart of Break Statement



Flowchart of break statement

```java
//Use of break in for loop
//Program will terminate when i equals 5

public class exampleofbreak {
    public static void main(String[] args) {

        for (int i=1; i<=10;i++){

            if(i==5){
                break;
            }
            System.out.println(i);

        }
    }
}
```

```java
//Use of break in while loop
//Program will terminate when i equals 5

public class BreakWhileExample {
public static void main(String[] args) {
    int i=1;
    while(i<=10){
        if(i==5){

            break;
        }
        System.out.println(i);
        i++;
    }
}
}
```

```java
//Use of break in do-while loop
//Program will terminate when i equals 5

public class BreakDoWhileExample {
public static void main(String[] args) {
    int i=1;
    do{
        if(i==5){

            break;
        }
        System.out.println(i);
        i++;
    }while(i<=10);
}}
```

Java Continue Statement

- The continue statement is used in loop control structure when you need to jump to the next iteration of the loop immediately.
- The Java *continue statement* is used to continue the loop.

- It continues the current flow of the program and skips the remaining code at the specified condition.
- We can use Java continue statement in all types of loops such as for loop, while loop and do-while loop.

```java
//Use of Continue in for loop
//Program will terminate when i equals 5

public class ContinueExample {
public static void main(String[] args) {
    for(int i=1;i<=10;i++){
        if(i==5){
            continue;
        }
        System.out.println(i);
    }
}
}
```

```java
//Use of Continue in while loop
//Program will terminate when i equals 5

public class ContinueWhileExample {
public static void main(String[] args) {
    int i=1;
    while(i<=10){
        if(i==5){
            i++;

            continue;
        }
        System.out.println(i);
        i++;
    }
}
}
```

```java
//Use of Continue in do-while loop
//Program will terminate when i equals 5

public class ContinueDoWhileExample {
public static void main(String[] args) {
    int i=1;
    do{
        if(i==5){
                i++;
            continue;
        }
        System.out.println(i);
        i++;
    }while(i<=10);
}
}
```

پوهنتون کاردان
KARDAN UNIVERSITY

Write a program that can find the average of 7 subjects, if any subject mark is less than 60, it should skip to the next?

Questions ?

پوهنتون کاردان
KARDAN UNIVERSITY

Thank You…!