Day 4

Thu 21 Aug 2017
Java Tutorial
Instructor: Hameed Mahmoud

Day 4

Basics of Java Programming Language

Objectives

1. Practice
2. Switch
3. For loop
4. While loop
5. Break and continue

Java Switch Statement

The Java *switch statement* executes one statement from multiple conditions. It is like if-else-if ladder statement.

Syntax:

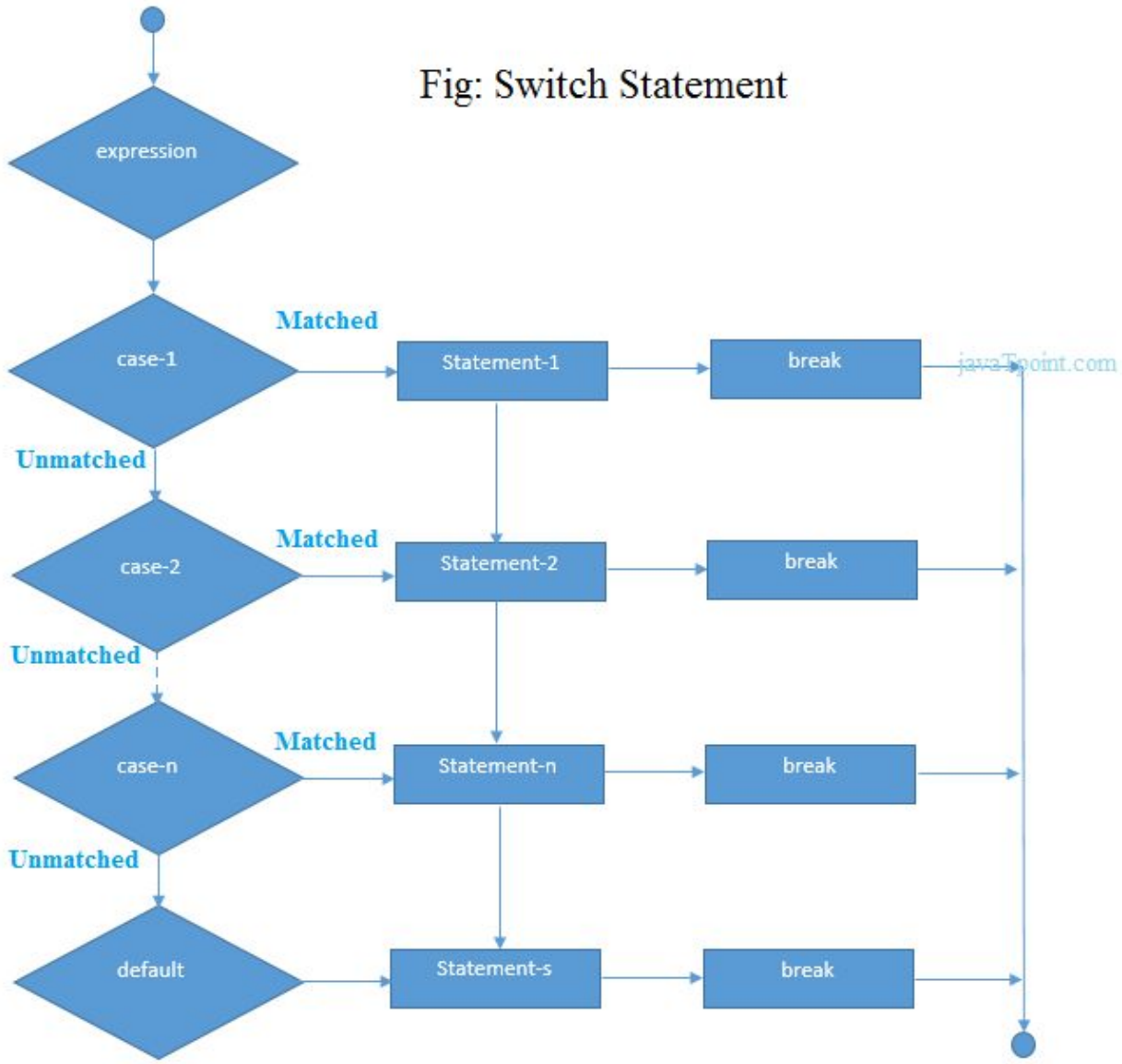
```
switch(expression){

case value1:

 //code to be executed;

 break;  //optional

case value2:

 //code to be executed;

 break;  //optional
```

.....

default:

code to be executed if all cases are not matched;

}



Example:

```
public class SwitchExample {

public static void main(String[] args) {

    int number=20;

    switch(number){

    case 10: System.out.println("10");break;

    case 20: System.out.println("20");break;

    case 30: System.out.println("30");break;

    default:System.out.println("Not in 10, 20 or 30");

    }

}

}
```

Output:

20

Java For Loop

The Java *for loop* is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop.

There are three types of for loop in java.

- Simple For Loop
- For-each or Enhanced For Loop
- Labeled For Loop

Java Simple For Loop

The simple for loop is same as C/C++. We can initialize variable, check condition and increment/decrement value.

Syntax:

```
for(initialization;condition;incr/decr){

//code to be executed

}
```

Example:

```
public class ForExample {

public static void main(String[] args) {

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

        System.out.println(i);

    }

}

}
```

Output:

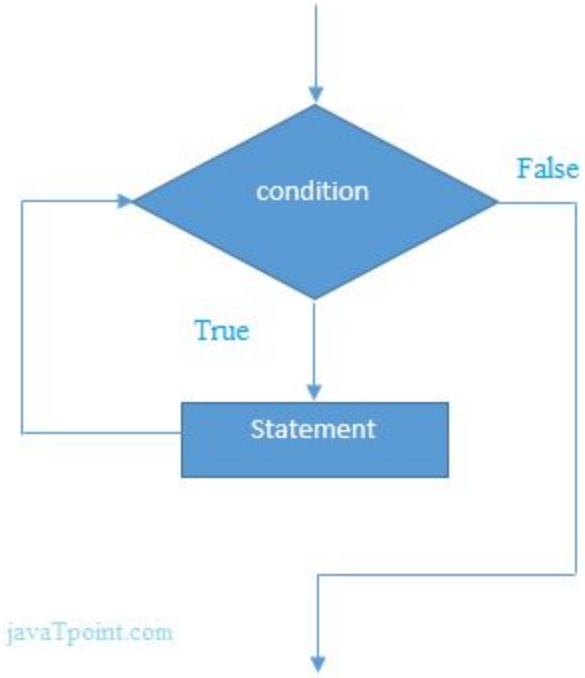
```
1
2
3
4
5
6
7
8
9
10
```

Java While Loop

The Java *while loop* is used to iterate a part of the program several times. If the number of iteration is not fixed, it is recommended to use while loop.

Syntax:

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



Example:

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

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

```
   1.  }
```

Output:

```
1
2
3
4
5
6
7
8
9
10
```

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:

```
public class WhileExample2 {
```

```
public static void main(String[] args) {
    while(true){
        System.out.println("infinitive while loop");
    }
}
}
```

Output:

```
infinitive while loop
infinitive while loop
infinitive while loop
infinitive while loop
infinitive while loop
ctrl+c
```

Java Break Statement

The Java *break* is used to break loop or switch statement. It breaks the current flow of the program at specified condition. In case of inner loop, it breaks only inner loop

Java Break Statement with Loop

Example:

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

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

Output:

```
1
2
3
4
```

Java Continue Statement

The Java *continue statement* is used to continue loop. It continues the current flow of the program and skips the remaining code at specified condition. In case of inner loop, it continues only inner loop.

Syntax:

```
1. jump-statement;
2. continue;
```

Java Continue Statement Example

Example:

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

```
    for(int i=1;i<=10;i++){
        if(i==5){
            continue;
        }
        System.out.println(i);
    }
}
}
```

Output:

```
1
2
3
4
6
7
8
9
10
```