Symbolic Constant

Type Casting

Java Operators and Expression

JAVA control statements

Control Statements are used to control the execution flow of the program. With the help of control statements, we can control the order of execution of the program, based on logic and the values.

In Java, control statements can be divided into the following three categories:

- Selection Statements
- Iteration Statements
- Jump Statements

Selection statements:

Selection statements allow us to control the flow of program execution, on the basis of the outcome of an expression or state of a variable, known during runtime.

Selection statements can be divided into the following categories:

- The if statements
- The if-else statements
- The switch statements

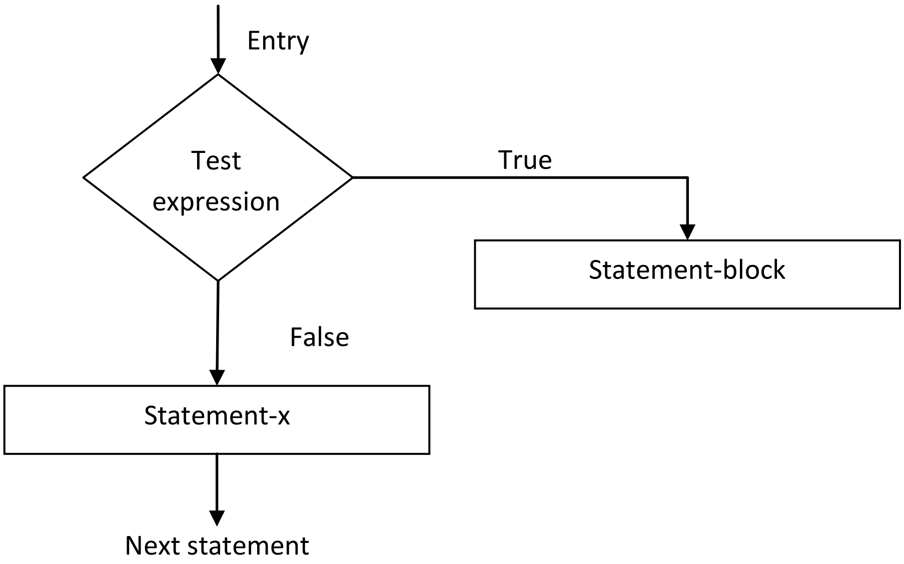
If statement

The general form of a **if** statement is

```
If (test expression)
{
        Statement-block;
}
Statement-x;
```

The '*statement-block'* may be a single statement or a group of statements. If the *test-expression* is true, the '*statement-block'* will be executed, otherwise the *statement-block'* will be skipped and the execution will jump to the '*statement-x'*.

When the condition is true both the statement-block and statement-x are executed in sequence as shown below:

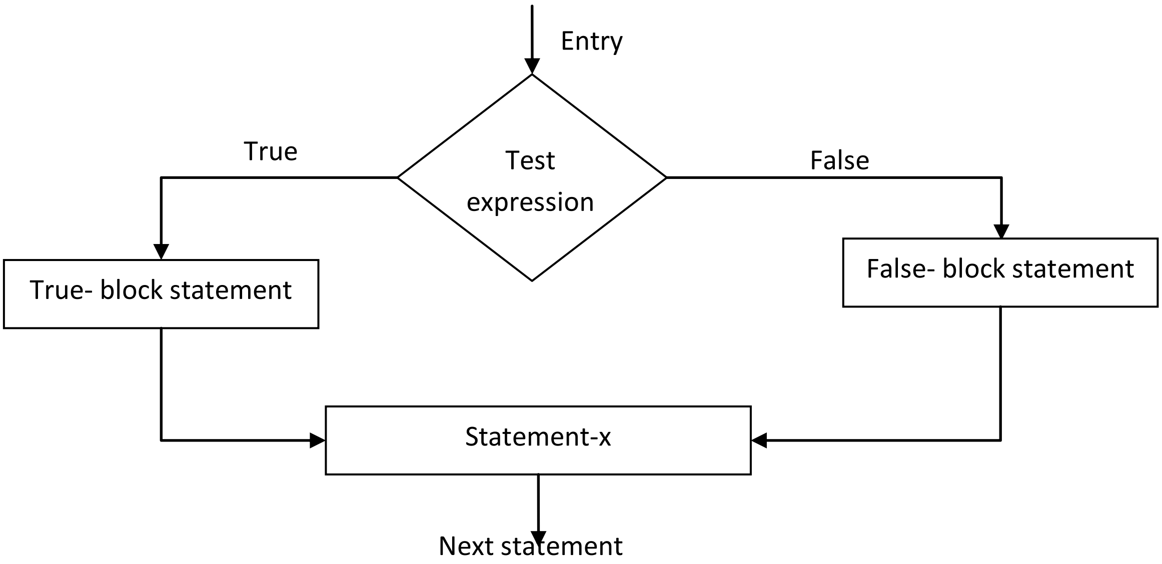


The If…Else statement

The if…else statement is an extension of the simple if statement. The general form is:

```
If (test expression)
{
        True-block Statement(s);
}
else
{
        False-block Statemen(s);
}
Statement-x;
```

If the test expression is true, then the *True-block statement(s)* are executed, otherwise, *false-block statement(s)* are executed. In either case, either *True-block or false-block* will be executed, not both.



The Switch Statement

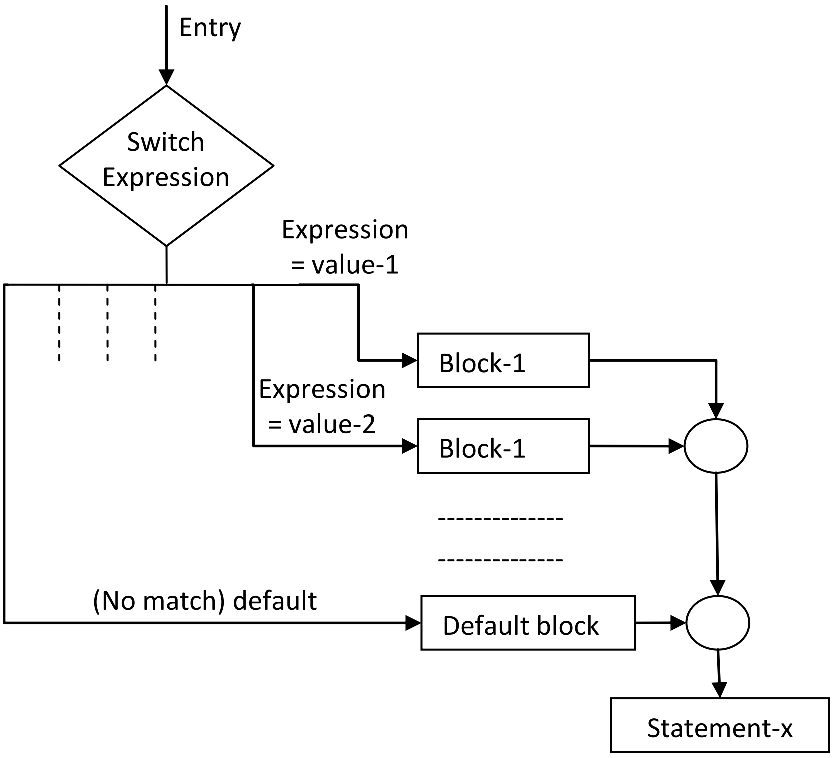
The switch statement tests the value of a given variable (or expression) against a list of **case** values and when a match is found a block of statements associated with that **case** is executed. The general form of the **switch** statement is as shown below:

```
        Switch( expression)
        {
                Case value-1:
                        Block-1
                        Break;
                Case value-2:
                        Block-2
                        Break;
                ……..
                ……..
                Default:
                        Default-block
                        Break;
}
Statement-x;
```

The ‘expression is an integer expression or characters.
Value-1, value-2, ... are the constants or constant expressions and are known as case labels. Each of these values should be unique within a **switch** statement,
Block-1, block-2, ..., are statement lists and may contain zero or more statements.

When the **switch** is executed, the value of the expression is successively compared against the values *value-1, value-2, ... If a* ***case*** is found whose value matches with the value of the expression then the block of statements that follows the case are executed.

The **break** statement at the end of each block signals the end of a particular case and cause an exit from the **switch** statement, transferring the control to the *statement-x* following the **switch.**


Entry
Switch Expression
Expression = value-1
Block-1
Expression = value-2
Block-1

(No match) default
Default block
Statement-x