

2. Agenda

- “ Relation Operator
- “ Conditional logic
- “ Basic Looping
- “ Arrays
- “ For each loop
- “ The Switch Statement

Relation Operator

	Operator	Integer Floating Point Example	Character Example	Boolean Example
Greater than	>	5>4	'c' > 'a'	Not applicable
Greater than or equal to	>=	5>=4 5>=5	'c' >= 'a' 'c' >= 'c'	Not applicable
Less than	<	4<5	'a' < 'c'	Not applicable
Less than or equal to	<=	4<= 5 4<=4	'a' <= 'c' 'a' <= 'a'	Not applicable
Equal to	==	5 == 5	'a' == 'a'	True == True False == False
Not equal to	!=	4 != 5	'a' != 'a'	True != False False != Ture

Conditional Assignment

“ Assign a value to a variable based on the result of a condition.

. result = condition ? true-value : false-value

```
Example: 1  int v1 = 7;
            int v2 = 5;
            int vMax = v1 > v2 ? v1 : v2;
            System.out.println(vMax);           //o/p -> ?
```

```
Example: 2  float bread = 30;
            float guest = 4;
            float breadPerGuest = guest == bread ? bread : bread/guest;
            System.out.println(breadPerGuest);   //o/p -> ?
```

If-else Statement

“ An if statement conditionally executes a statement

```
if(condition)
    true-statement;
else
    false-statement;
```

“ Please note here, else statement is an optional statement. We can write/omit as per our program requirement.

“ Example: We will write a program to understand this type of looping.

Chaining if-else statement

“ If-else statements chained together are evaluated in order until one is true. If nothing is true, it will go to Else

```
if(condition-1)
    true-statement-1;
else if(condition-2)
    true-statement-2;
.
.
.
else if(condition-N)
    true-statement-N;
else
    false-statement;
```

Block Statement

“ A block statement groups statement into a compound statement.

```
If (some-condition){  
    statement-1;  
    statement-2;  
    .  
    .  
    statement-n;  
}
```

“ We will take an example and then it will be more clear. Calculator example is a perfect one.

Nested if-statement

- “ Inside a if-statement, we can open N number of another if-else statement.
- “ Why it is required ?
- “ We may have some scenario that for one particular conditional statement, we have to check, extra condition. Ex. In case of calculator program, for division we should check if denominator is 0 or not.
- “ How we will do it, we will take a look.

Block Statement and variable Scope

“ A variable declared within a block is not visible outside the block.

- . A variable's range of visibility is known as the variable's scope.
- . We can understand this concept using previous example.

```
float bread = 30;
```

```
float guest = 4;
```

```
If (guest >0) {
```

```
    float breadPerGuest = bread/guest;
```

```
}
```

```
System.out.println(breadPerGuest); //Compilation error
```

- . How to fix this error, we have either use print statement inside the block or declare the variable outside the block.

Logical Operator with if-else

	Operator	What resolves to true
AND	&	true & true
Or		false true, true false, true true
Exclusive or (XOR)	^	false^true, true^false
Negation	!	false

We will see this with an example of finding highest number in a given 3 number.

Conditional Logical Operator

	Operator	What resolves to True
Conditional and	&&	true && true
Conditional or		False true, true --

- “ Resolves following conceptually similar rules as non-conditional and/or
- “ Only execute the right-side if needed to determine the result.
 - . && only execute right-side if left side is true.
 - . || only execute right-side if left side is false

Why Conditional logical Operator

” There are some places where conditional logical operator is very much useful. Example:

```
float bread = 30;
```

```
float guest = 4;
```

```
If (guest >0 & bread/guest > 10) {
```

```
    System.out.println(“Heavy lunch”);
```

```
}
```

```
    System.out.println(“**Success**”);
```

```
/*in case of & operator it will throw an error, but in case  
if case of conditional logical operator it will not*/
```

Making Calculator Program better

- “ We will try to make the calculator program better whatever we have studied so far.
- “ In case of division operation, there is a possibility that we input denominator as 0 which is an invalid scenario.
- “ As a better programmer, we should check such kind of scenario.
- “ We will see in the code how to do it.

Loops in Java

- “ Three basic ways to write loops in Java.
 - . While loop
 - . Do-While loop
 - . For loop

While Loop

- “ Repeatedly executes a statements as long as the condition is true.
 - . Here condition is checked at the start of loop
 - . There is a possibility that statement may never execute at all if condition is not satisfied.
 - . Syntax is
 - `while(some condition)`
 - `Statement;`
 - . We will see a program to make it more clear.

Do-While loop

“ Repeatedly execute a statements as long as the condition is true.

- . Condition is checked at the end of the loop
- . So even if the condition is not true in the starting itself, it will run the statement at least one time.
- . Syntax is

do

 statement;

 while(some condition);

- . We will see a program to make it more clear.

For Loop

- “ Repeatedly executes a statement as long as the condition is true.
 - . Condition checked at the loop start
 - . Provides simplified notation for loop control values
 - . Syntax
 - for(initialize, condition, update)
 - Statements;
 - . We will see a program to make it more clear. (Program to printing 1-10)

Array

- “ Provides an ordered collection of elements. Mechanism to store similar more than one value of similar data type.
 - . Each element accessed via an index
 - . Index range from 0 to number of elements minus 1
 - . Number of elements can be found via array's length value

Array Contd..

“ Syntax to define Array

```
float[] varName = new float[3];
```

“ It is declared to store 3 variable of float data type.

“ It will 1st allocate a memory space inside your computer

0

1

2



“ We will see a program to make it more clear.

For each loop

“ Executes a statements once for each member in an array

- . Handles getting collection length
- . Handles accessing each value
- . Syntax

```
For(loop-variable-declaration: array)  
statements;
```

Switch

- “ Transfer controls to a statements based on a value
 - . Simplifies testing against multiple possible matches
 - . Only primitive types supported as char and integers
 - . A match can execute more than one statements
 - . Use break to avoid “falling through”
 - . Can optionally include default to handle any unmatched values

Switch contd..

” Syntax:

```
switch(test-value){  
    case value-1:  
        statements;  
    case value-2:  
        statements;  
    .  
    .  
    .  
    case value-n:  
        statements;  
    default:  
        statements;
```

Summary

- “ Use the if-else statement to provide conditional logic
 - . If-else statement can be chained together
- “ Block statements use brackets to group statements
 - . Variables declared within a block are not visible outside the block
- “ Both while and do-while loops execute as long as condition is true.
 - . The do-while loop body always execute at least once.
- “ The for loop provides simplified notation for loop initialization and control.
- “ For-each statement handles details of executing once for each array member.
- “ Switch statement simplifies notation of testing against multiple matches.