Lecture 2 Introduction to Basic Operators Java

Contents

Author

Waheed Rafiq

Postgraduate with a BSc(Hons) 1st in Computer Science from Birmingham City University

I.T Software Developer for QA

Disclaimer

How to use this guide

These notes and YouTube videos are designed to be dyslexic friendly; there are few symbols develop to aid the student. Anyone can use this technique, but if you are dyslexic, then it's a must.

Symbol	*Meaning*
	*'Speak out loud.'* Enables the learner to understand the concept/code/subject being discuss.
	*'Flash Card'* *You are encouraged to make notes and understand the concept.*
	*'Assessment question.'* *Ensure you understand the topic as this will be in the assessment.*
	*'Code practice.'* *As a rule of thumb, never COPY & PASTE code* • *Write out the code from memory* • *Repeat the code using different scenarios*
	*'Brain Peak'* *To increase mental awareness, the recommendation is for ten mins before proceeding with the lecture play a maths game 'any level'. Which will aid you with better concentration at the topic covered?*
Highlighted words	*Any words highlighted implies you to do a bit of research by using Google or Bing search engine. You could be assessed at a later stage of the lectures.*

Previous Lecture

Declare datatypes for Personal details fields?

In the previous lecture video, we learnt what variables are. Before proceeding on to the next tutorial, we need to ensure that we understand the basic please test yourself by clicking on this link https://testmoz.com/1306521.

The answer to the practical task is as follows:

```
// Declare variables  note string datatype is in uppercase
    String fname = "Waheed"; // first name
    String lastname = "Rafiq"; // last name
    short age = 18; // I wish ....
    int houseNumber = 33;
    char gender = 'M';
    double wage = 300.568;
    System.out.println("----------------------------");
    System.out.println("    Personal Details         ");
    System.out.println("----------------------------");
    System.out.println("");

    System.out.println("Your name is: " + fname + " " + lastname);
    System.out.println("You are : " + age + " years old :)  ");
    System.out.println("Your gender is : " + gender);
    System.out.println("You earn £ " + wage);
```

The above is an example of how you should have solved the practical task that was asked of you in Lecture 1.

Basic Operators in Java?

Java is one of many languages that provides a good set of operators to manipulate variables. Emphasising that these 'Operators' should only be used when required just like a toolbox full of tools. Java operators are categorised into six groups.

1)Arithmetic Operators

2) Bitwise Operators

3) Logical Operators

4) Relational Operators

5) Assignment Operators

6) Misc. Operators

The best way to study these lectures is to watch YouTube video and used these notes as a reference guide.

1) Arithmetic Operators are used in mathematical expressions.

Operators	Description	Example
+ (Addition)	Adds values on either side of the operator	Num1 + Num2 = 10
-(Subtraction)	Subtracts right-hand operand from left-hand operand.	Num1 – Num2 = -10
*(Multiplication)	Multiplies values on either side of the operator	Num1 * Num2 = 100
/ (Division)	Divides left-hand operand by right-hand operand	Num1 / Num2 = 4
%(Modulus)	Divides left-hand operand by right-hand operand and returns the remainder.	Num2 % Num1 = 0
++(Increment)	Increase the value of operand by 1	Num1 ++ = 2
--(Decrement)	Decreases the value of operand by 1	Num1 -- = 1

The Bitwise Operators

Bitwise operators can be applied to the integer, long, int, short, char and byte. All of which we have covered in lecture 1.

Bitwise operators work on bits and perform bit-by-bit operations.

In this example Num1 = 20 and Num2 = 13

Num1 in binary is = 00010100

Num2 in binary is = 00010100

Operator	Description
& (bitwise and)	Binary AND Operator copies a bit to the result if it exists in both operands
| (bitwise or)	Binary OR Operator copies a bit if it exists in either operand
^ (bitwise XOR)	Binary XOR Operator copies the bit if it is set in one operand but not both
~ (bitwise compliment)	Binary One Complement Operator is unary and has the effect of 'flipping' bits.
<< (left shift)	Binary Left Shift operator. The left operands value is moved right by the number of bits specified by the right operand.
>> (right shift)	Binary right shift operator. The left operands value is moved right by the number of bits specified by the right operand.
>>> (Zero fill right shift)	Shift right Zero fill operator. The left operands value is moved right by the number of bits specified by the right operand and shifted values are filled up with zeros.

Logical Operators

Logical operators are the most commonly used in programming.

Let say M = true and F= false

Operator	Description	Example
&&(logical and)	If both condition are true	M && F is false
||(logical OR)	If any condition is true	M ||F is true
!(logical not)	Refer to NOT operator use to reverse the logical states of its operand. If a condition is true then Logical NOT operator will make it false.	! (M && F) is true

Relational Operators

Relational Operators as the term suggest is anything to do with the relationship between data types, variables and conditions which will be cover in the next lecture. This operator tests or defines the relation between two entities which include numerical equality example 10 = 10 and inequalities 5 > 2.

Number1= 5 ; Number2 = 10

Operator	Description	Syntax
== (equal to)	Is it the same value if it is then the condition is true	Number1 == Number 2is not true
! = (not equal to)	Check if the value is not same if it's NOT equal then condition become true.	Number1! = Number2 is true
> (greater than)	Check if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	Number2 > Number1 is true
< (less than)	Check if the value of left operand is less than the value of right operand if it is then conditioned becomes true	Number1 < Number2 so its true
>= (greater than or equal to)	Check if the value of the left operand is greater than or equal to the value on the right if yes condition is true	Number1 >= number2 is false
<= less than or equal to	Opposite of the above	Number1 <= Number2 is true.

The Assignment Operators

Assignment Operators are the most common operators that you will be using daily, already we have used in lecture one the 'equal operator' = which assigns the value of its right operand to its left operand.

Operator	Description
=	Assignment operator right side operands to left side operand.
+=	Add AND assignment operator. It adds right operand to the left operand and assigns the result to the left operand.
-=	Subtract AND assignment operator. It subtracts right operand from the left operand and assigns the result to the left operand.
*=	Multiply AND assignment operator. It multiplies right operand with the left operand assigns the result to the left operand.
/=	Divide AND operator. Divides left operand with the right operand and assign the result to the left operand.
%=	Modulus AND assignment operator. Takes modulus using two operators and assign the result to the left operand.
<<=	Left shift AND assignment operator.
>>=	Right shift AND assignment operator.
&=	Bitwise AND assignment operator
^=	Bitwise exclusive OR and assignment operator
|=	Bitwise inclusive OR and assignment operator.

Miscellaneous Operator

Java also supports conditional operator '?' which is also referred as 'ternary operator.'

This operator consists of three operands and is used to evaluate Boolean expressions.

To use some of this operator we have to cover one conditional statement that is known as the 'if statement.'

The 'if statement' implies "if the specified condition is true then execute the following…" this can be extended by else, and if else more will be discussed in lecture 3, for now, the following syntax is used. Example Num1 = 5, num2 = 10

```
If(num1 < numb 2)

{

  // do something here …
```

}

Coding Basic Operators

“Now we understand what Basic operators are let’s put theory into practice.”

Few points to note in our IDE we came across word ‘package’ this is a namespace that organises a set of related classes and interfaces, one way to look at this is root folder that has different folders within such as images and scripts, etc...

Tips: useful link here to work out Mods if you don’t know how to do them

https://www.easycalculation.com/algebra/modulo-calculator.php

The programming task is to complete the missing code in lecture 2 YouTube video

Once you have filled in the gaps in the code return and clicked on this link

testmoz.com/1311043 [good luck]