CCS0023/L

Object Oriented Programming (Java)

Operators

03

LESSON #3

Operators

JAVA OPERATORS

Unary Arithmetic Operators
Binary Arithmetic Operators
Bitwise Operators
Shift Operators
Relational and Logical Operators
Cast Operator
Assignment Operator
Ternary Operator

Unary Arithmetic Operators

+ -	Sign Operators
++	Increment
--	Decrement

Increment and decrement operators have two formats: postfix and prefix.

- Operators

Binary Arithmetic Operators

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo Division

Precedence Level
* / %
+ -

Arithmetic Operators

There are some general guidelines to consider when creating arithmetic expressions:

- ➢ Use parentheses to alter precedence.
- ➢ Consider the size of resulting values and possible loss of precision.
- ➢ Apply arithmetic operators’ precedence level.

Arithmetic Operators

- ➢When using negative numbers with modulus calculation, drop the negative signs from either operand and calculate the result. The sign of the left operand is the sign of the result.

- ➢ The (+) operator can be used to add numbers and String objects.

- ➢ Dividing integer by zero results in the throwing of ArithmeticException

Bitwise Operators

- works on integer values, by manipulating its bit-pattern equivalent.

Operator	Use	Operation
~	~Op	**Bitwise inversion (unary)**
|	Op1 | Op2	**Bitwise OR**
&	Op1 & Op2	**Bitwise AND**
^	Op1 ^ Op2	**Bitwise EX-OR**

OR logic table

op1	op2	result
0	0	0
0	1	1
1	0	1
1	1	1

AND logic table

op1	op2	result
0	0	0
0	1	0
1	0	0
1	1	1

Shift Operators

- performs bit manipulation on data by shifting the bits of its first operand right or left. This table summarizes the shift operators available in the Java programming language.

Operator	Use	Operation
>>	op1 >> op2	shift bits of op1 right by distance op2
<<	op1 << op2	shift bits of op1 left by distance op2
>>>	op1 >>> op2	shift bits of op1 right by distance op2 (unsigned)

Relational Operators

- compares two values and determines the relationship between them.

Operator	Use	Returns true if
>	op1 > op2	op1 is greater than op2
>=	op1 >= op2	op1 is greater than or equal to op2
<	op1 < op2	op1 is less than op2
<=	op1 <= op2	op1 is less than or equal to op2
==	op1 == op2	op1 and op2 are equal
!=	op1 != op2	op1 and op2 are not equal

Logical Operators

➢ often used with relational operators to construct more complex decision-making expressions.

Operator	Use	Operation
&&	Op1 && Op2	Short circuit AND
||	Op1 || Op2	Short circuit OR
&	Op1 & Op2	AND
|	Op1 | Op2	OR

Cast Operators

- used in converting (casting) one type into another type or an object into another object.

  Syntax:

  ```
  (Cast type) Value;
  ```

  Example:

  ```
  double dbl = 55.66;
  int number = (int)dbl;
  ```

Assignment Operators

- used in converting (casting) one type into another type or an object into another object.

Syntax:

```
(Cast type) Value;
```

Example:

```
double dbl = 55.66;
int number = (int)dbl;
```

Ternary Operator

➢ an operator that deals with three operands.

Syntax:

```
variable = condition ? value : value;
```

Example:

```
int x = 20, y = 30;
int biggerNumber = 0;
biggerNumber = x>y ? x : y;
// What do u think will be the value of
the biggerNumber??
```

THANK YOU

FEU INSTITUTE OF TECHNOLOGY
1992