Republic of the Philippines

Department of Education

REGION III

SCHOOLS DIVISION OFFICE OF NUEVA ECIJA

LEARNING ACTIVITY SHEET

SPECIAL PROGRAM IN ICT

BASIC PROGRAMMING 8

Third Quarter, Week 1

EXPRESSIONS AND OPERATORS

Background Information for Learners

Java provides a rich set of operators to manipulate variables. We can divide all the Java operators into the following groups:

- Arithmetic Operators
- Relational Operators
- Bitwise Operators
- Logical Operators
- Assignment Operators
- Misc Operators

The Arithmetic Operators

Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra. The following table lists the arithmetic operators.

Assume integer variable A holds 10 and variable B holds 20, then

Operator	Description	Example
+ (Addition)	Adds values on either side of the operator.	A + B will give 30
- (Subtraction)	Subtracts right-hand operand from left-hand operand.	A - B will give -10
* (Multiplication)	Multiplies values on either side of the operator.	A * B will give 200
/ (Division)	Divides left-hand operand by right-hand operand.	B / A will give 2
% (Modulus)	Divides left-hand operand by right-hand operand and returns remainder.	B % A will give 0
++ (Increment)	Increases the value of operand by 1.	B++ gives 21
-- (Decrement)	Decreases the value of operand by 1.	B-- gives 19

The following program is a simple example which demonstrates the arithmetic operators.

```
public class Test
{
    public static void main(String args[])
{
      int a = 10;
      int b = 20;
      int c = 25;
      int d = 25;
      System.out.println("a + b = " + (a + b) );
      System.out.println("a - b = " + (a - b) );
      System.out.println("a * b = " + (a * b) );
      System.out.println("b / a = " + (b / a) );
      System.out.println("b % a = " + (b % a) );
      System.out.println("c % a = " + (c % a) );
      System.out.println("a++   = " +  (a++) );
      System.out.println("a-- = " + (a--) );

      // Check the difference in d++ and ++d
      System.out.println("d++   = " +  (d++) );
      System.out.println("++d   = " +  (++d) );
   }
}
```

This will produce the following result

```
a + b = 30
a - b = -10
a * b = 200
b / a = 2 b
% a = 0 c %
a = 5
a++   = 10
b--   = 11
d++   = 25
++d   = 27
```

The Relational Operators

There are following relational operators supported by Java language.
Assume variable A holds 10 and variable B holds 20, then

Operator	Description	Example
== (equal to)	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(A == B) is not true.
!= (not equal to)	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(A != B) is true.
> (greater than)	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.
< (less than)	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.
>= (greater than or equal to)	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<= (less than or equal to)	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.

The following program is a simple example that demonstrates the relational operators.

```
public class Test {

   public static void main(String args[])
{        int a = 10;        int b = 20;

      System.out.println("a == b = " + (a == b) );
      System.out.println("a != b = " + (a != b) );
      System.out.println("a > b = " + (a > b) );
      System.out.println("a < b = " + (a < b) );
      System.out.println("b >= a = " + (b >= a) );
System.out.println("b <= a = " + (b <= a) );
   }
}
```

This will produce the following result

```
a== b = false                                      a < b = true
a!= b = true                                       b >= a = true
a > b = false                                      a<= a = false
```

The Logical Operators

The following table lists the logical operators:

Assume Boolean variables A holds true and variable B holds false, then

Operator	Description	Example
&& (logical and)	Called Logical AND operator. If both the operands are nonzero, then the condition becomes true.	(A && B) is false
|| (logical or)	Called Logical OR Operator. If any of the two operands are non-zero, then the condition becomes true.	(A || B) is true
! (logical not)	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is true

The following simple example program demonstrates the logical operators.

```
public class Test
{
  public static void main(String args[])
  {
      boolean a = true;
      boolean b = false;

      System.out.println("a && b = " + (a&&b));
      System.out.println("a || b = " + (a||b) );
      System.out.println("!(a && b) = " + !(a && b));
   }
}
```

This will produce the following result

```
a && b = false a
|| b = true !(a
&& b) = true
```

The Assignment Operators

Following are the assignment operators supported by Java language.

Operator	Description	Example
=	Simple assignment operator. Assigns values from right side operands to left side operand.	C = A + B will assign value of A + B into C
+=	Add AND assignment operator. It adds right operand to the left operand and assign the result to left operand.	C + = A is equivalent to C = C + A
-=	Subtract AND assignment operator. It subtracts right operand from the left operand and assign the result to left operand.	C - = A is equivalent to C = C – A
*=	Multiply AND assignment operator. It multiplies right operand with the left operand and assign the result to left operand.	C * = A is equivalent to C = C * A
/=	Divide AND assignment operator. It divides left operand with the right operand and assign the result to left operand.	C / = A is equivalent to C = C / A
%=	Modulus AND assignment operator. It takes modulus using two operands and assign the result to left operand.	C % = A is equivalent to C = C % A
<<=	Left shift AND assignment operator.	C << = 2 is same as C = C << 2
>>=	Right shift AND assignment operator.	C >> = 2 is same as C = C >> 2
&=	Bitwise AND assignment operator.	C &= 2 is same as C = C & 2
^=	bitwise exclusive OR and assignment operator.	C ^= 2 is same as C = C ^ 2
|=	bitwise inclusive OR and assignment operator.	C |= 2 is same as C = C | 2

Learning Competency with Code

Use expressions and operators in a Java program

Exercises/Activities

ACTIVITY 1: Answer the attached Google Form in our Google Classroom

ACTIVITY 2: Check the image of a given code in our Google Classroom and write its result in the Private Comment Section

REFLECTION: Type your answer in the Private Comment section of our Google Classroom

- On your own opinion, why is it important to know the different functions of each operator?

References for Learners

https://www.tutorialspoint.com/java/java_basic_operators.htm
https://www.tutorialspoint.com/java/java_arithmatic_operators_examples.htm
https://www.tutorialspoint.com/java/java_logical_operators_examples.htm
https://www.tutorialspoint.com/java/java_assignment_operators_examples.htm
https://www.tutorialspoint.com/java/java_relational_operators_examples.htm

Prepared by: **NOEMI F. MAGNO**
Name of Writer

Noted by: **LABERNE A. LADIGNON**
Division ICT Coordinator/ OIC EPS

Activity 1

Direction: Identify the operator being described.

____________________ 1. Increases the value of operand by 1.
____________________ 2. Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.
____________________ 3. If both the operands are non-zero, then the condition becomes true.
___________________4. Assigns values from right side operands to left side operand.
___________________5. Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.
___________________6. Checks if the values of two operands are equal or not, if yes then condition becomes true.
___________________7. Decreases the value of operand by 1.
___________________8. Multiplies values on either side of the operator.
___________________9. Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.
___________________10. If any of the two operands are non-zero, then the condition becomes true.

Activity 2

```
public class Test {

   public static void main(String args[])
{        int a = 5;        int b = 10;
int c = 15;        int d = 30;

      System.out.println("a + b = " + (a + b) );
      System.out.println("b - a = " + (b - a) );
      System.out.println("c * d = " + (c * d) );
      System.out.println("b / a = " + (b / a) );
      System.out.println("b % a = " + (b % a) );
      System.out.println("c % a = " + (c % a) );
      System.out.println("a++   = " +  (a++) );
      System.out.println("a-- = " + (a--) );

      // Check the difference in d++ and ++d
      System.out.println("d++   = " +  (d++) );
      System.out.println("++d   = " +  (++d) );
   }
}
```