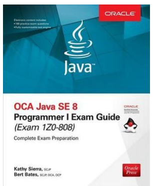


ORACLE

Java

OCA Java SE 8 Programmer I Exam Guide (Exam 1Z0-808)

Complete Exam Preparation

Kathy Sierra, Bert Bates

Chapter 4

Operators

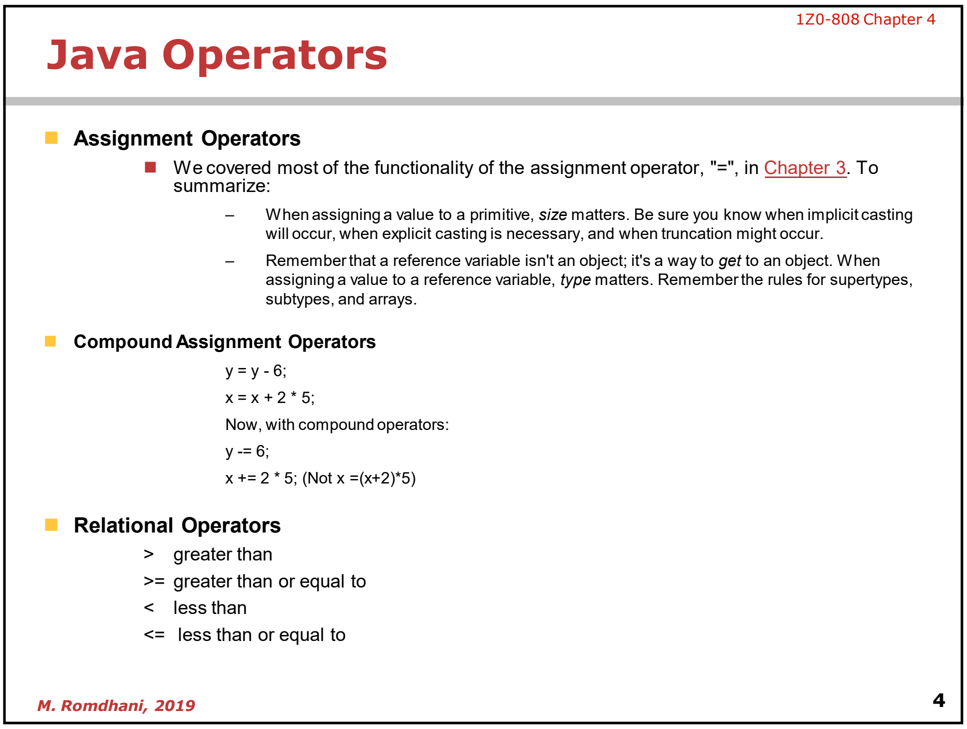
Business Training

Content

- **Java Operators** (OCA Objectives 3.1, 3.2, and 3.3)

Java Operators

(OCA Objectives 3.1, 3.2, and 3.3)



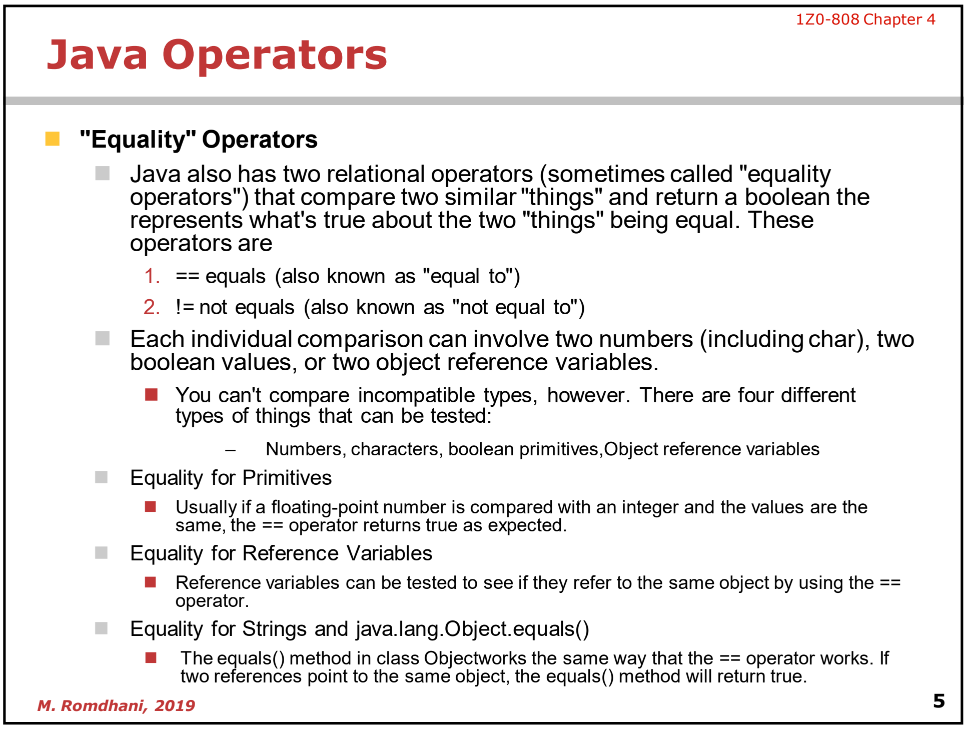
Java Operators

- **Assignment Operators**
  - We covered most of the functionality of the assignment operator, "=", in Chapter 3. To summarize:
    - When assigning a value to a primitive, *size* matters. Be sure you know when implicit casting will occur, when explicit casting is necessary, and when truncation might occur.
    - Remember that a reference variable isn't an object; it's a way to *get* to an object. When assigning a value to a reference variable, *type* matters. Remember the rules for supertypes, subtypes, and arrays.
- **Compound Assignment Operators**

```
y = y - 6;
x = x + 2 * 5;
Now, with compound operators:
y -= 6;
x += 2 * 5; (Not x =(x+2)*5)
```

- **Relational Operators**

>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

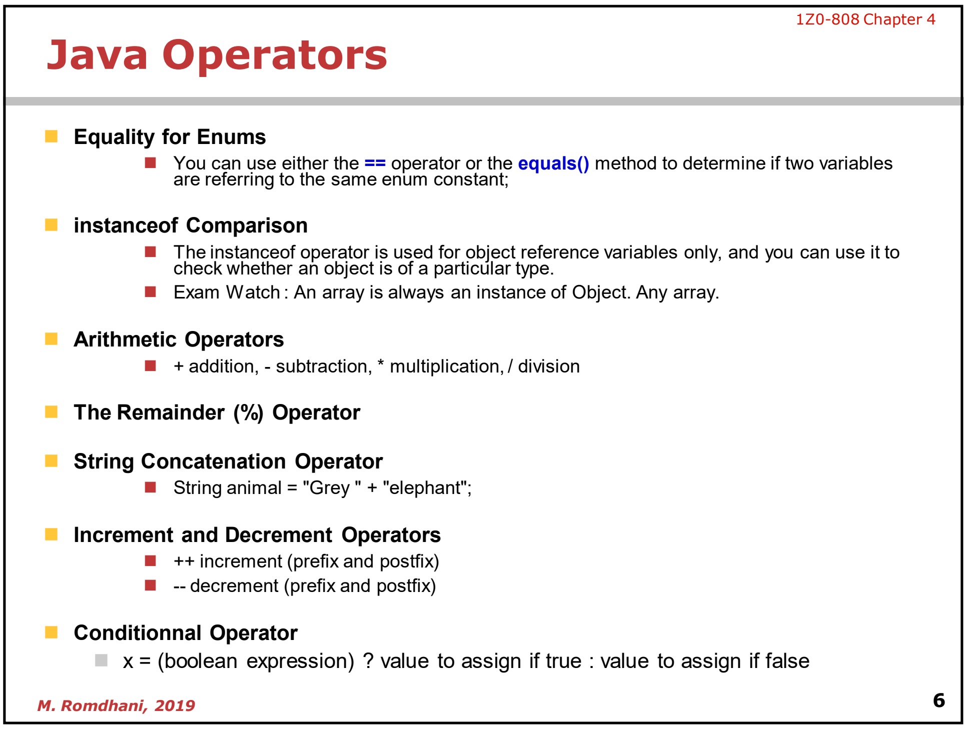


1Z0-808 Chapter 4

Java Operators

- **"Equality" Operators**
  - Java also has two relational operators (sometimes called "equality operators") that compare two similar "things" and return a boolean the represents what's true about the two "things" being equal. These operators are
    1. == equals (also known as "equal to")
    2. != not equals (also known as "not equal to")
  - Each individual comparison can involve two numbers (including char), two boolean values, or two object reference variables.
    - You can't compare incompatible types, however. There are four different types of things that can be tested:
      - Numbers, characters, boolean primitives,Object reference variables
  - Equality for Primitives
    - Usually if a floating-point number is compared with an integer and the values are the same, the == operator returns true as expected.
  - Equality for Reference Variables
    - Reference variables can be tested to see if they refer to the same object by using the == operator.
  - Equality for Strings and java.lang.Object.equals()
    - The equals() method in class Objectworks the same way that the == operator works. If two references point to the same object, the equals() method will return true.

M. Romdhani, 2019 5

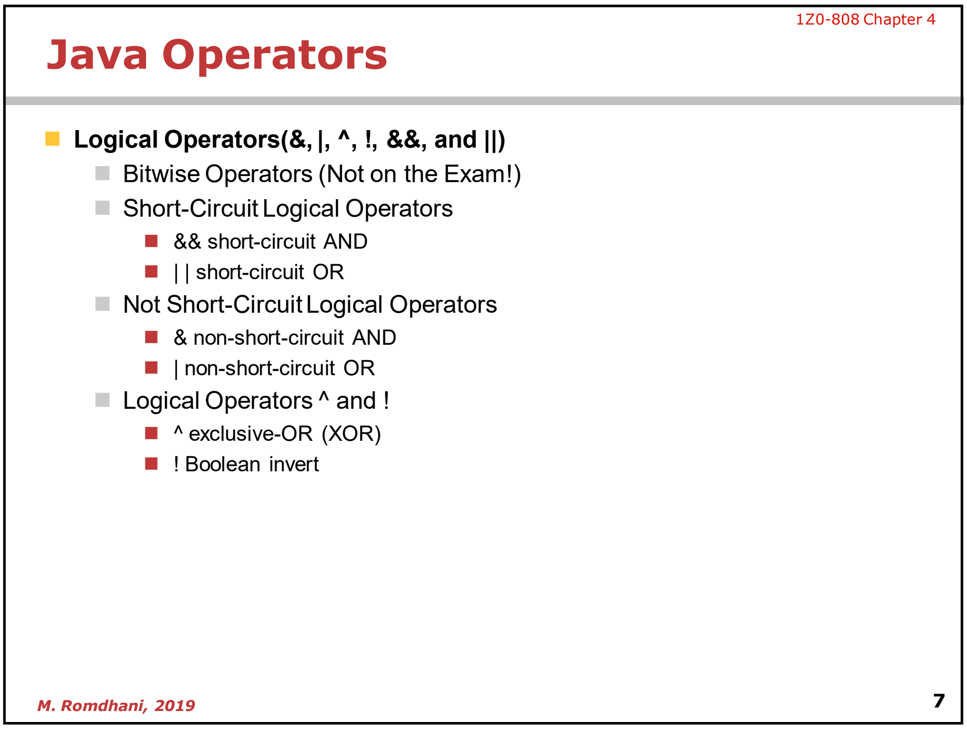


1Z0-808 Chapter 4

Java Operators

- **Equality for Enums**
  - You can use either the **==** operator or the **equals()** method to determine if two variables are referring to the same enum constant;
- **instanceof Comparison**
  - The instanceof operator is used for object reference variables only, and you can use it to check whether an object is of a particular type.
  - Exam Watch : An array is always an instance of Object. Any array.
- **Arithmetic Operators**
  - + addition, - subtraction, * multiplication, / division
- **The Remainder (%) Operator**
- **String Concatenation Operator**
  - String animal = "Grey " + "elephant";
- **Increment and Decrement Operators**
  - ++ increment (prefix and postfix)
  - -- decrement (prefix and postfix)
- **Conditionnal Operator**
  - x = (boolean expression) ? value to assign if true : value to assign if false

M. Romdhani, 2019 6



1Z0-808 Chapter 4

Java Operators

- **Logical Operators(&, |, ^, !, &&, and ||)**
  - Bitwise Operators (Not on the Exam!)
  - Short-Circuit Logical Operators
    - && short-circuit AND
    - | | short-circuit OR
  - Not Short-Circuit Logical Operators
    - & non-short-circuit AND
    - | non-short-circuit OR
  - Logical Operators ^ and !
    - ^ exclusive-OR (XOR)
    - ! Boolean invert

M. Romdhani, 2019

7