OOP Labsheet-3

Prof.R.Gururaj

Java Operators

- ❑ Arithmetic
- ❑ Bitwise
- ❑ Relational
- ❑ Logical

Arithmetic operators:

Operator	Result
+	Addition (also unary plus)
–	Subtraction (also unary minus)
*	Multiplication
/	Division
%	Modulus
++	Increment
+=	Addition assignment
–=	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment
––	Decrement

Note:

- operands must be of numeric type
- cannot use them on boolean types
- can use them on char types

Bitwise operators:

Operator	Result
~	Bitwise unary NOT
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
>>	Shift right
>>>	Shift right zero fill
<<	Shift left
&=	Bitwise AND assignment
|=	Bitwise OR assignment
^=	Bitwise exclusive OR assignment
>>=	Shift right assignment
>>>=	Shift right zero fill assignment
<<=	Shift left assignment

Can be applied to long, int, short, char and byte

Act upon individual bits of the operands

Practice problem-1:
// A program demonstrating Bitwise operators in java

Relational operators

Operator	Result
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Outcome is a boolean value

Most frequently used in the expressions that control the if statement and the various loop statements

Boolean Logical operators

Operator	Result
&	Logical AND
|	Logical OR
^	Logical XOR (exclusive OR)
||	Short-circuit OR
&&	Short-circuit AND
!	Logical unary NOT
&=	AND assignment
|=	OR assignment
^=	XOR assignment
==	Equal to
!=	Not equal to
?:	Ternary if-then-else

Operate on boolean operands.

Combine two boolean values to form a resultant boolean value.

short-circuit AND (&&) – also called conditional AND

short-circuit OR (||) – also called conditional OR

Practice problem-2:

// A program demonstrating Logical operators in java

Assignment and ternary operator

int x, y, z;

x = y = z = 100; //chain of assignments

expression1 ? *expression2* : *expression3*

- *expression1* evaluates to a boolean value
- both *expression2* and *expression3* are required to return the same or compatible type

Operator precedence

Highest						
++ (postfix)	– – (postfix)					
++ (prefix)	– – (prefix)	~	!	+ (unary)	– (unary)	(*type-cast*)
*	/	%				
+	–					
>>	>>>	<<				
>	>=	<	<=	instanceof		
==	!=					
&						
^						
|						
&&						
||						
?:						
->						
=	op=					
Lowest						

Practice problem-3:

// A program demonstrating operator precedence in java

Practice problem-4

Write a Java program that will create a 2-D array of integers (row-0: IDs of students, row-1: marks) with following data.

Row-0: 101 103 105 106 109

Row-1: 67 70 43 89 56

Index [0] [1] [2] [3] [4]

Marks of student with ID 101 are 67

Marks of student with ID 106 are 89 so on..

After that, write a loop where you will ask to input ID from user through keyboard and then search for the entry in row-1 for the given ID, if exists get the corresponding marks from row-2, and will print the marks, if ID is not a valid, print some error message and go for asking the input for ID. Exit loop if ID if -999 is typed.

// Students will try this on their own before one possible solution is shown

Practice Problem-5

Java code to do the following.

1. Define a class Student with id(int), name (String), age(int), and marks which is an array of three integer elements to store marks for three subjects(initially set to zero), as attributes.
2. Student class will have a three argument constructor to initialize the attribute values.
3. Define a method setMarks() which takes three integers and assign them to elements in marks[] array.
4. Write studentDetails() to print id, name , age and marks of student.
5. Write a class StudentDemo that creates two Student objects with data- 105, “Raj”, 21, set marks{67, 90, 65}; 109, “Gopal”,23, set marks{70, 75, 55}. And finally print details.

// Students will try this on their own before one possible solution is shown

*****