ESc101: Fundamental of Computing

I Semester 2008-09

Lecture 4

I Primitive Data Types in JAVA, Operators, Expression evaluation

II Block of code

(Primitive) Data types in JAVA

Domain	Java type
Integer	byte, short, int, long
Fractional numbers	float, double
Boolean	boolean
Characters	char

Arithmetic operators		
Operator	Meaning	Result
+	addition	Integer
_	subtraction	"
*	multiplication	u
/	integer division	íí
%	mod	íí.

19/4:?

19%4:?

Arithmetic operators		
Operator	Meaning	Result
+	addition	Integer
_	subtraction	"
*	multiplication	u
/	integer division	íí
%	mod	íí.

19/4:4

19%4:3

Relational operators		
<,	less than	Boolean
<=	less than or equal	"
==	equal	"
! =	not equal	ιι

2 < 3:?

3 == 3 : ?

Relational operators		
<,	less than	Boolean
<=	less than or equal	"
==	equal	"
! =	not equal	"

2 < 3: true

3 == 3: true

In a similar fashion >, >= are defined.

Operation defined on Data types for fractional numbers

Same as that of integer data type except:

/ is the same as the usual division operator.

% is the remainder by usual division.

19.0/4.0:4.75

 $19.0\%4.0:\ 3.0$

Operation defined on Data type for Boolean

Logical operators			
Operator	Meaning	Result	
!	NOT	boolean	
&,&&	AND	boolean	
,	OR	boolean	
Relational operators			
==	Equal	boolean	
! =	Not equal	boolean	

Evaluation of Expressions

2+3*4 is equal to ??

96/4/2 is equal to ??

Is 3/2 * 60 * 60 equal to 60 * 60 * 3/2?

Evaluation of Expressions

2+3*4 is equal to **14**

96/4/2 is equal to 12

Is 3/2*60*60 equal to *60*60*3/2 ? : **NO**

Evaluation of Expressions

An important tool:

It is always better to use parentheses in writing any expression.

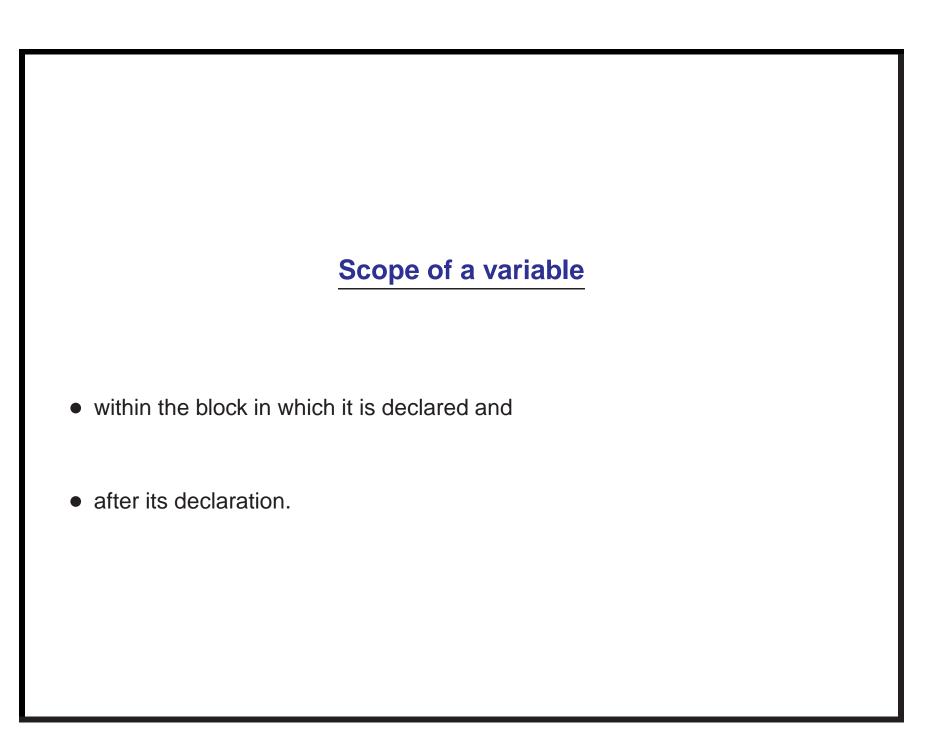
However, if the expression is not fully parenthesized, then the following rules are followed

- terms in parentheses are evaluated first
- the operators of higher precedence are evaluated <u>before</u> the operators of lower precedence.
- two consecutive operators have same precedence, they are evaluated from left to right.(called left associative).
- +,-,*,/,% are left associative.

II: Block of code

Definition : a sequence of statements enclosed between $\{$ and $\}$.

```
For example
{
   Statement1;
   Statement2;
   .
   .
   Statementk;
}
```



Example: Scope of vatiable

```
1.class scope
2.{
      public static void main(String args[])
3.
4.
          int i;
          i = 100;
5.
6.
          System.out.println("value of i here is "+i);
7.
               int j;
8.
               j=55;
9.
               i = i*j;
10.
11.
               System.out.println(i);
12.
               System.out.println(j;)
13.
14.
          System.out.println(j);
15.
16.}
```

The above code will give copilation error at line 14 because no j exists at this line.

The scope of i is from line 5 to 14, scope of j is from line 9 to 12 only.

Motivation for If statement

Find the minimum of two or more numbers.

```
class if_example
 public static void main(String args[])
    int i,j,max;
    //---write code here so that max
    //---stores the bigger of i and j
```

In next lecture, we shall introduce If statement.