

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	<code>byte, short, int, long</code>
Fractional numbers	<code>float, double</code>
Boolean	<code>boolean</code>
Characters	<code>char</code>

Operation defined on Data types for Integer

Arithmetic operators		
Operator	Meaning	Result
+	addition	Integer
—	subtraction	“
*	multiplication	“
/	integer division	“
%	mod	“

19/4 : ?

19%4 : ?

Operation defined on Data types for Integer

Arithmetic operators		
Operator	Meaning	Result
+	addition	Integer
—	subtraction	“
*	multiplication	“
/	integer division	“
%	mod	“

19/4 : 4

19%4 : 3

Operation defined on Data types for Integer

Relational operators		
<	less than	Boolean
<=	less than or equal	“
==	equal	“
!=	not equal	“

$2 < 3 : ?$

$3 == 3 : ?$

Operation defined on Data types for Integer

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 before 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;  
}
```

Scope of a variable

- within the block in which it is declared and
- after its declaration.

Example : Scope of variable

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

The above code will give compilation 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.