

CISC124 – Expressions & Loops		
<ul style="list-style-type: none"> ▪ Expressions <ul style="list-style-type: none"> • Elements • Variables and literals • Keywords • Operators ▪ Loops <ul style="list-style-type: none"> • While loop • Example 		
Winter 2019	F. de la Parra	1

•1

Elements of Expressions		
<p>An expression in Java can consist of :</p> <ul style="list-style-type: none"> • Variables • Literal values • Keywords • Operators • Method invocations • Punctuation <p>And evaluates to a value used for some purpose</p> <p>Examples: (totalCount < 50) evaluates to true/false</p> <p>Sum = Sum + 50; evaluates to a number</p>		
Winter 2019	F. de la Parra	2

•2

Variables and Literals		
<ul style="list-style-type: none"> • Variables: have names and types <ul style="list-style-type: none"> • Store a value (primitive types) or a reference to an object (class types) • Camel case naming convention → roomNumber • Static typing → set when declaring a variable (Example: int count;) • Literal values <ul style="list-style-type: none"> • Numbers → 5 5.25 3.14E-2F others • Characters → 'a' '8' '\u0049' • Strings → "Hello" 		
Winter 2019	F. de la Parra	3

•3

Keywords		
<p>About 53 keywords in the Java language</p> <ul style="list-style-type: none"> • Cannot be used to name variables, classes, objects, etc. • Case sensitive and must comply with java syntax rules • Primitive types: byte, short, int, long, float, double, char, boolean • Conditionals: if, else, switch, case, break, default • Loops: while, for, do, break, continue • Classes and Methods: class, interface, package, public, private, protected, static, void ... • Exceptions: try, catch, finally, throw, throws 		
Winter 2019	F. de la Parra	4

•4

Operators		
<p>They operate on one, two or three expressions to return a value! There exists an order of evaluation for operators (precedence).</p>		
<ul style="list-style-type: none"> • Unary operators <ul style="list-style-type: none"> • Operator (Expression) • Binary operators <ul style="list-style-type: none"> • (Expression 1) Operator (Expression 2) • Conditional operator (ternary operator) <ul style="list-style-type: none"> • (Expression 1) ? (Expression 2) : (Expression 3) 		
Winter 2019	F. de la Parra	5

•5

Unary Operators		
<ul style="list-style-type: none"> • Arithmetic sign change: - • Boolean negation: ! • Pre and post-increment (integer): ++ • Pre and post-decrement (integer): -- • Bitwise complement (byte): ~ 		
Winter 2019	F. de la Parra	6

•6

Binary Operators

- Boolean operators:
 - `&&` (conditional AND) → operates on true/false expr.
 - `||` (conditional OR) → operates on true/false expr.
 - `&` (boolean AND) → operates on true/false expr, and as a bitwise operator on integers
 - `|` (boolean OR) → operates on true/false expr, and as a bitwise operator on integers
 - `^` (boolean XOR) → operates on true/false expr, and as a bitwise operator on integers

Winter 2019

F. de la Parra

7

•7

Binary Operators (continued)

- Arithmetic: `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division), `%` (modulo)
 - `9/4` evaluates to 2
 - `9/4.0f` evaluates to 2.5
 - `9%4` evaluates to 1
- String concatenation: `+`
 - `"Hello" + " world"` → `"Hello world"`
- Comparison: `==` (equals), `!=` (not equals), `<` (less than), `<=` (less than or equal), `>` (greater than), `>=` (greater than or equal),

Winter 2019

F. de la Parra

8

•8

Binary Operators (continued)

- Assignment operators:
 - `=` (assign right-side expression to left-side-expression)
 - It is right associative → `a = b = c` operates as `a = (b = c)`
- Combination assignment operators. The general form is:


```
var op= value
```

 - Arithmetic operators plus assignment: `+=`, `-=`, `*=`, `/=`, `%=`
 - Bitwise operators plus assignment: `&=`, `|=`, `^=`

Winter 2019

F. de la Parra

9

•9

Binary Operators (continued)

- The `instanceof` operator checks whether a value is an instance of class or not:
 - `"Hello" instanceof String` returns `true`
 - `"Hi" instanceof String` returns `true` (strings are also instances of `Object`)
 - `Null instanceof String` returns `false` (null is never an instance of anything)

Winter 2019

F. de la Parra

10

•10

Conditional Operator

- Conditional operator (ternary operator)
 - `(Expression 1) ? (Expression 2) : (Expression 3)`
 - Evaluate `Expression1`
 - If true evaluate `Expression 2` and return its value
 - If false evaluate `Expression 3` and return its value

Winter 2019

F. de la Parra

11

•11

While loop

```
while (BooleanExpression) {
    Block 1 of Statements
}
```

```
while
loop
nesting
to any
level

while (BooleanExpression 1) {
    while (BooleanExpression 2) {
        Block of Statements
    }
}
```

Winter 2019

F. de la Parra

12

•12

While loop	
<pre>while (Expression) { Statements Block1 continue; Statements Block 2 }</pre>	Control returns to evaluate expression (Block 2 not executed)
<pre>while (Expression) { Statements Block1 break; Statements Block 2 }</pre>	While loop exits to previous nesting level (if one exists) (Block 2 not executed)
Winter 2019	F. de la Parra 13

•13