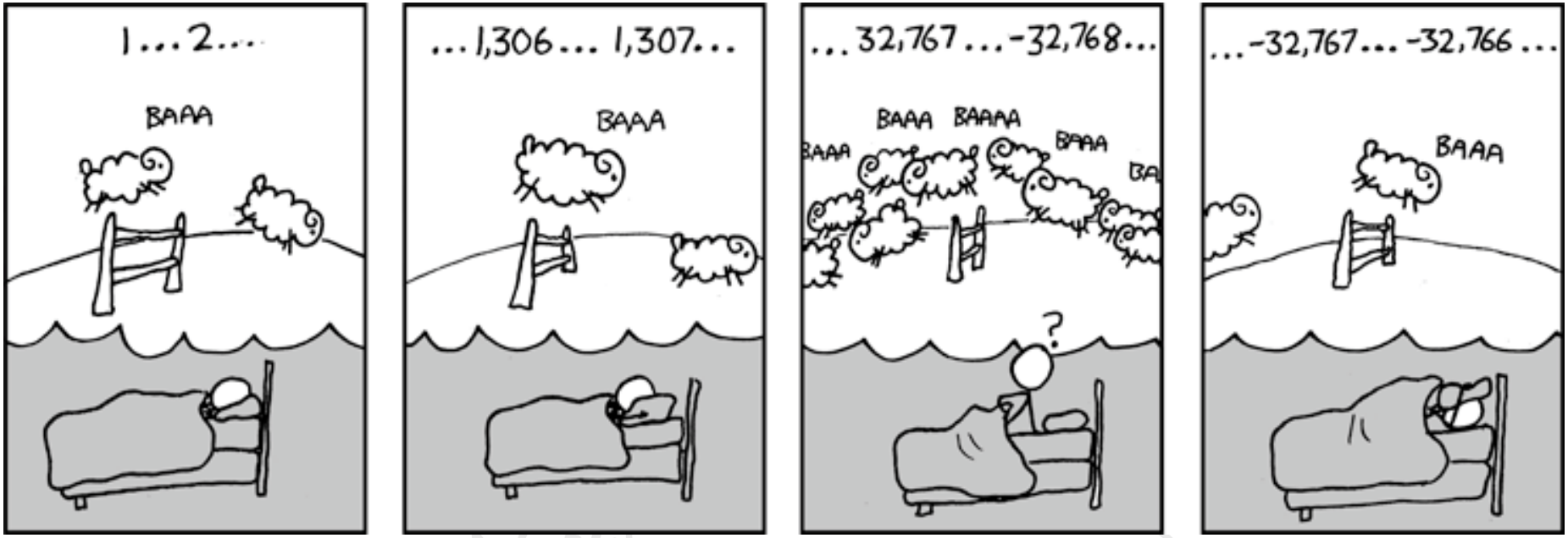

A programmer counting Sheep to put himself to sleep

CSN-103: Fundamentals of Object Oriented Programming

Operators in Java

- Divided into four groups
  - Arithmetic
  - Relational
  - Logical
  - Bitwise

- Arithmetic

Operands must be of a numeric or char type

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

Operators in Java

- Arithmetic Compound Assignment Operators

a = a + 4	equivalent to	a += 4
a = a - 4	equivalent to	a -= 4
a = a * 4	equivalent to	a *= 4
a = a / 4	equivalent to	a /= 4
a = a % 4	equivalent to	a %= 4

- Increment and Decrement Operators

x = x + 1	equivalent to	x++	or	++x
x = x - 1	equivalent to	x--	or	--x

Postfix vs. Prefix Form

Postfix vs. Prefix Form

- ++ and -- in expressions
  - Prefix: Operand is incremented/decremented before the value is obtained for use in the expression
  - Postfix: Previous value is obtained for use in the expression, and then the operand is incremented/decremented

```
x = 42;
x= ++x;
```

```
x = 42;
x = x+1;
y = x;
```

Equivalent

```
x = 42;
y = x++;
```

```
x = 42;
y = x;
x = x+1;
```

Equivalent

Relational Operators

- *Relational* operators determine the relationship (equality and ordering) that one operand has to the other

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

- The outcome of these operations is a **boolean** value
- Equality test is done using == NOT = which is Assignment operator
- Inequality test is done using !=

Boolean Logical Operators

- Operate only on **boolean** operands

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

Boolean Logical Operators

- Results of different each Boolean logical operation

A	B	A | B	A & B	A ^ B	!A
False	False	False	False	False	True
True	False	True	False	True	False
False	True	True	False	True	True
True	True	True	True	False	False

- Short-Circuit Logical Operators
  - OR operator results in **true** when A is true, no matter what B is
  - AND operator results in **false** when A is false, no matter what B is
- Use || and && instead of | and &
  - Java will not evaluate the right-hand operand when the outcome of the expression can be determined by the left operand alone

The Assignment and Ternary Operator

- Assignment
  - Different than == (Equal to) operator
  - Create a chain of assignments

```
int x, y, z;
x = y = z = 100;
```

- The Ternary (?) Operator
  - The ? has this general form:

```
expression1 ? expression2 : expression3
```

  - *expression1* can be any expression that evaluates to a boolean value.
  - If *expression1* is true, then *expression2* is evaluated; otherwise, *expression3* is evaluated

System and Scanner Classes

Output to the Monitor

- System class
  - A class in the *java.lang* package
  - System class is a collection of methods and variables
    - Standard input
    - Standard **output**
    - Standard error

- Standard **output**

```
System.out.println(data)          // Move to next line
System.out.print(data)
```

Input from the Keyboard

- Scanner class
  - Reads formatted input and converts it into its binary form
  - Can read from **Keyboard**, File, and String
- Steps

```
import java.util.Scanner;                    //Necessary
Scanner sc = new Scanner(System.in);


byte b = sc.nextByte();
short s = sc.nextShort();
int a = sc.nextInt();
long b = sc.nextLong();
double c = sc.nextDouble();
float f = sc.nextFloat();
boolean b = sc.nextBoolean();
String s = sc.nextLine();
```

nextChar() does not exist