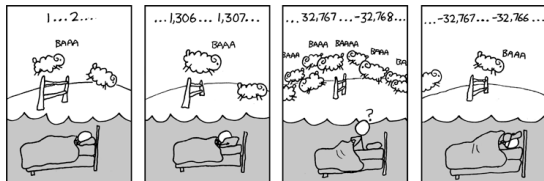


Reading for this lecture and previous lecture:

Sections 1.1, 1.2, 1.3. Lab 1 will give you practice with concepts and details of 1.2, 1.3.

PLive: Lesson 0, Lesson page 1.3, Activity 1-4.1.



Results of last lecture's questionnaire

Number completed: 305 (about 50 students did not)

No programming experience: 171 (56%)

< 5 months experience: 73 (24%)

>= 5 months experience: 61 (20%)

Number not in science/engineering: 44 (14%)

Number expressing concern because of lack of experience compared to others or the work load: 88 (29%)

Terminology

Programming language (Java, C, Fortran, Matlab, Python): a language in which you write programs, often to be executed on a computer.

Program: A set of instructions, written in a programming language, to be executed (carried out, performed) to get some task done. Like a recipe in a cookbook.

Machine language. The language of instructions that a computer is able to execute (carry out, perform).

Java Compiler. A program that translates a Java program into a machine language form so that it can be executed on a computer.

Type: A set of values together with operations on them.

Memorize this definition!

Write it down several times.

Type **integer**:

values: ..., -3, -2, -1, 0, 1, 2, 3, 4, 5, ...

operations: +, -, *, /, unary -

Type **int**:

$-2^{31} .. 2^{31}-1$

values: -2147483648, -2147483647, ..., -3, -2, -1, 0, 1, 2, 3, 4, 5, ..., 2147483646, 2147483647

operations: +, -, *, /, unary -

Principal: These **int** operations must yield an **int**.

b / c rounds toward 0

Integer.MIN_VALUE Integer.MAX_VALUE

Type: A set of values together with operations on them.

Type **double**:

values: Examples: $-22.51E6$ equivalent to -22510000 or $-22.51 * 10^6$
 $22.51E-6$ equivalent to $.00002251$ or $22.51 * 10^{-6}$

An approximation to the real numbers.

operations: +, -, *, /, unary -

Double.MIN_VALUE 4.9E-324

Double.MAX_VALUE 1.7976931348623157E308

Smallest POSITIVE value

Casting between double and int

(double) 2 casts 2 to type **double**. Value is 2.0
 This is a *widening* cast, and it may be done automatically by Java if it is necessary.

(int) 2.56 casts 2.56 to type **int**. Value is 2
 This is a *narrowing* cast. Java *never* does it automatically because it might lose information and it may be done automatically by Java if it is necessary.

Other examples:

(double)(int) 2.56 Value is 2.0

(double) 2.56 Value is 2.56

Type: A set of values together with operations on them.

Type **boolean**:

values: **true false**

operations: ! (not) && (and) || (or)

!b is true if b is false and false if b is true

b && c, read "b and c",
is **true** if both b and c are **true**, false otherwise

b || c, read "b || c",
is **true** if b is true or c is **true**, false otherwise **==, not = !!**

i < j i <= j i == j i >= j i > j i != j
evaluate to **true** or **false**

7

Precedence of operators (page 23)

- Unary operators: + - !
- Binary arithmetic: * / %
- Binary arithmetic: + -
- Arithmetic relations: < > <= >=
- Equality relations: == !=
- Logical and: &&
- Logical or: ||

You will practice working with expressions in Lab 01.

8

Variables, p. 26

- A variable is a name together with a value.
- A variable is a named box with a value in the box.

Memorize definition!

Write it down several times.

x 5 **int**

Here's variable x, with value 5. It can contain an **int** value.

area 20.1 **double**

Here's variable area, with value 20.1. It can contain a **double** value.

9

Declaration of a variable, p. 26

In Java, a *declaration of a variable* gives the name of the variable and the type of value it can contain.

Memorize these two definitions!

Write them down several times.

int x;

Here's a declaration of x, indicating that it can contain an **int** value.

double area;

Here's a declaration of area, indicating that it can contain a **double** value.

Assignment statement, p. 27

Execution of an assignment statement stores a value in a variable.

To execute the assignment

<var> = <expr>;

evaluate expression <expr> and store its value in variable <var>.

x = x + 1; Evaluate expression x+1 and store its value in variable x.

10

Type: A set of values together with operations on them.

Type **String**: values: Any sequence of characters

operations: + (catenation, or concatenation)

String literal: sequence of chars enclosed in double quotes:
e.g. "abcx3\$g<&"

Here is **String** catenation: "bc" + "fg"

Question: what can you do to convert the value of an expr, e.g. 3.1459 + 4.1, to a **String**

+ is overloaded: Consider x + y

- (1) If one operand (x or y) is a **String**, the other is converted to a **String** (if necessary) and catenation is done.
- (2) Otherwise, if one operand is a **double**, the other is cast to a **double** (if necessary) and a **double** addition is done.
- (3) Otherwise, both operands are **ints** and an **int** addition is done,

Recitations (Labs) in the Engineering ACCEL LAB Start Today and Tomorrow!

To get to the ACCEL Lab, go into the Engineering Library in Carpenter Hall, walk straight until you come to a staircase on your left, and go up the stairs.

Do not be concerned if you haven't been able to register for a recitation section. Just go to the one you want this week.

Here are the times of the recitation-labs: Attend ONE of them.

Tuesday: 12:20, 1:25, 2:30, 3:35

Wednesday: 12:20, 1:25, 2:30, 3:35

You can help by bringing your own laptop

12