CS-259
Data Structures with Java
Java: Primitive Types

Instructor: **Joel Castellanos**
e-mail: joel@unm.edu
web: http://cs.unm.edu/~joel/

Course website:
http://cs.unm.edu/~joel/cs259/

8/31/2009

Monday, Aug 31

- Quizzes on reading from chapter 3.
- Java Types
- Random Numbers
- Assignment of Project 1: Roulette
- Review of Java Prime Number Program
- Start Lab 2 (Prime factors).
- Homework: Due Wednesday, Sept 2 Read Chapter 3
  - Strings (53-63)
  - Input and Output (63-71)
  - Control Flow (71-88)

2

Quiz 1-6: Case Sensitive

In computer science, *Case Sensitive* means that:

a) Objects are composed of cases.

b) Cases are composed of objects.

c) Objects with cases can be sorted.

d) Objects respond to cases.

e) “Main” is different from “main”.

3

Quiz 1-7: Strongly Typed

In computer science, a *Strongly Typed Language* is one in which:

a) Reserved words are displayed in a **bold** font.

b) Every variable must have a declared type.

c) Words are always used rather than symbols. For example, type “plus” rather than “+”.

d) Variable names are given descriptive names which are often long.

e) Some words are reserved and cannot be used for variable names.

4

Logical Operator &&: Truth Table

```
public static void main(String[] args)
{ System.out.println("false && false = " +
                     (false && false));
  System.out.println("true  && false = " +
                     (true && false));
  System.out.println("false && true  = " +
                     (false && true));
  System.out.println("true  && true  = " +
                     (true && true));
}
```

```
false && false = false
true  && false = false
false && true  = false
true  && true  = true
```

5

Logical Operator ||: Truth Table

```
public static void main(String[] args)
{ System.out.println("false || false = " +
                     (false || false));
  System.out.println("true  || false = " +
                     (true || false));
  System.out.println("false || true  = " +
                     (false || true));
  System.out.println("true  || true  = " +
                     (true || true));
}
```

```
false || false = false
true  || false = true
false || true  = true
true  || true  = true
```

6

“Short circuit” Evaluation

- The text book says this expression is “safe”.
- What does that mean and why is it safe?

```
if (x != 0 && 1/x > x + y)
```

7

Quiz 1-8: Operator Hierarchy

```
int x = 2;
int y = 5;
boolean a = x*y>x+y;
boolean b = y-x<=x && y-x<=y;
boolean c = y-x<=x || y-x<=y;
System.out.println(a+", "+b+", "+c);
```

Please, discuss with neighbor.

No Typing during the quiz!

What is the output of the above code?

a) true, false, true
b) true, true, true
c) true, false, false
d) true, true, false
e) false, true, false

8

Java's Primitive Types

`byte`:	8-bit, [-128, 127].
`short`:	16-bit, [-32,768, 32,767] .
`int`:	32-bit, [-2,147,483,648, 2,147,483,647].
`long`:	64-bit, [-9,223,372,036,854,775,808, 9,223,372,036,854,775,807].
`float`:	32-bit,[1.4×10^{-45}, 3.4028235×10^{38}]
`double`:	64-bit, [4.9×10^{-324}, $1.7976931348623157 \times 10^{308}$]
`boolean`:	Only two possible values: **`true`** and **`false`**.
`char`:	16-bit, ['**`\u0000`**' (0), '**`\uffff`**' (65,535)].

9

byte

- The **`byte`** data type is an 8-bit signed two's complement integer.
- Minimum value of -128 and a maximum value of 127 (inclusive).
- The **`byte`** data type can be useful for saving memory in ***large*** arrays, where the memory savings actually matters.
- They can also be used in place of **`int`** where their limits help to clarify your code; the fact that a variable's range is limited can serve as a form of documentation.
- Default value: **`0`** or **`0x00`**

Hexadecimal Literal

10

short

- The **`short`** data type is a 16-bit signed two's complement integer.
- Minimum value of -32,768 and a maximum value of 32,767 (inclusive).
- Use a short to save memory in ***large*** arrays, where the memory savings actually matters.
- Default value: **`0`**
- Short.MAX_VALUE, Short.MAX_VALUE
  - **`short`** is a primitive type (with a lower-case s).
  - **`Short`** is an Object (with an upper-case S).

`public static finial` fields

11

int

- The **`int`** data type is a 32-bit signed two's complement integer.
- Minimum value of -2,147,483,648 and a maximum value of 2,147,483,647 (inclusive).
- For integral values, this data type is generally the default choice unless there is a reason to choose something else.
- This data type will most likely be large enough for the numbers your program will use, but if you need a wider range of values, use long instead.
- Default value: **`0`**
- **`Integer.MAX_VALUE, Integer.MIN_VALUE`**

12

long

- The **`long`** data type is a 64-bit signed two's complement integer.
- Minimum value of -9,223,372,036,854,775,808 and a maximum value of 9,223,372,036,854,775,807 (inclusive).
- Use this data type when you need a range of values wider than those provided by **`int`**.
- Default value: **`0L`**

A **`long`** literal must have the suffix: **`L`**
Lower case L also works, but it is bad because it looks too much like a one.

- **`Long.MAX_VALUE, Long.MIN_VALUE`**

13

float

- The **`float`** data type is a single-precision 32-bit IEEE 754 floating point.
- 6-7 significant decimal digits, [1.4×10^{-45}, 3.4028235×10^{38}]
- Use a float (instead of double) if you need to save memory in large arrays of floating point numbers.
  - This data type should never be used for precise values, such as currency. The java.math.BigDecimal class provides arbitrary-precision.
  - Some numbers which are expressed exactly in base 10 with 2 decimal places, are repeating, non-terminating in binary.
- Default value: **`0f`** , **`0F`**

14

double

- The double data type is a double-precision 64-bit IEEE 754 floating point.
- 15 significant decimal digits, [4.9×10^{-324}, $1.7976931348623157 \times 10^{308}$],
- For decimal values, this data type is generally the default choice.
- A double has twice the number of bits as a float, yet more than twice the number of significant digits. How is that possible?
- While a double is much more precise than a float, this data type should never be used for currency or other very precise values.
- Default value: `0`

15