COMP 110-001
Primitive and Class Types

Yi Hong

May 14, 2015

Review

- What are the two major parts of an object?
- What is the relationship between class and object?
- Design a simple class for Student
- How to use a variable?

Today

- Primitive type
  - Integer
  - Boolean
  - Float / Double
  - Character
- Class type

Data Types

- Class type: Object with both data and methods
  - Has the the same name as the class
  - Name begins with uppercase letter (recommended)
  - E.g.: Scanner, String, Student (user-defined)
- Primitive type: indecomposable values
  - Name begins with lowercase letters
  - E.g.: int, double, char, boolean, …
  - See Figure 2.1, p 52 for the full list

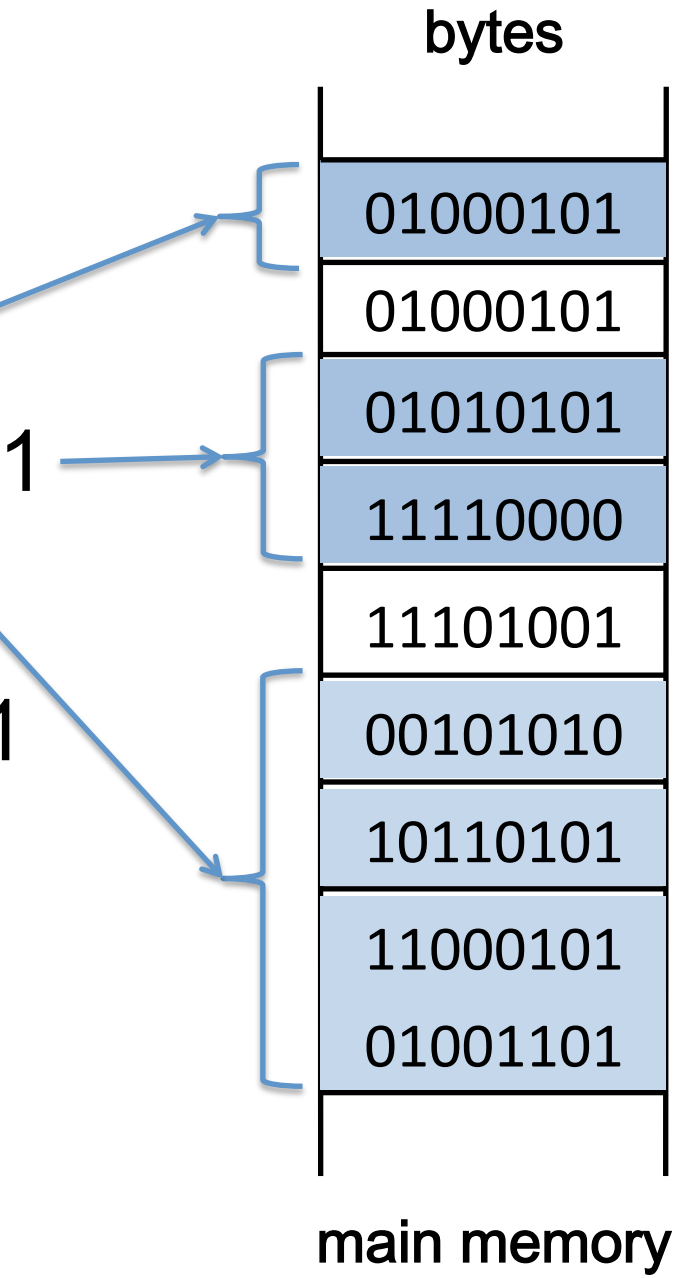
Primitive Types

- Integer (byte, short, int, long)
  - 0, -5, 10, 30
- Floating-point (float, double)
  - 0.5, -10.0, 12.98
- Single character (char)
  - A, c, %, S
- Boolean (boolean)
  - True or false

Integer

- byte: 1 byte, -2^7 to 2^7-1
- short: 2 bytes, -2^{15} to $2^{15}-1$
- int: 4 bytes, -2^{31} to $2^{31}-1$
- long: 8 bytes, -2^{63} to $2^{63}-1$

Numerical operations on integers return integers



Signed Conversions

- Signed binary to decimal, e.g., 10111101_2

1 0111101 Original value

0 1000010 Ones' complement

+1 Add 1

0 1000011 Result: 67

The sign is 1, a negative number, so $10111101_2 = -67_{10}$

- Signed decimal to binary, e.g., -102

102/2 = 51 rem. 0

51/2 = 25 rem. 1

25/2 = 12 rem. 1

12/2 = 6 rem. 0

6/2 = 3 rem. 0

3/2 = 1 rem. 1

1/2 = 0 rem. 1

1 1 0 0 1 1 0

0 1100110 : +102

1 0011010 Two's complement (ones' complement and add 1)

Floating-point

- Has a fractional part
  - E.g.: 5.0
  - float: 4 bytes, single-precision, smaller range, lower precision
  - double: 8 bytes, double-precision, larger range, higher precision

  If you cannot decide between the types float and double, use double

Single Character (Unicode)

- Char: 2 bytes, 0 to 2^{16}-1
- Single quotes enclose a character
  - E.g.: ‘a’, ‘A’
  - Uppercase letters and lowercase letters are different characters

Boolean

- boolean: 1 bit, true or false
- Boolean operators
  - && (and), || (or), ! (negation)

&&	true	false
true	true	false
false	false	false

||	true	false
true	true	true
false	true	false

! true	false
! false	true

Assignment Compatibilities

- Usually, we need to put values of a certain type into variables of the same type
- However, in some cases, the value will automatically be converted when types are different
- A value can be assigned to a variable whose type allows more precision
  - byte → short → int → long → float → double

```
int age;    age = 10;
double length;   length = age;   ✓
```

Type Casting

- Changes the data type of a value from its normal type to some other type
  - E.g: double distance = 9.0;

    int points = distance; ✗

    int points = (int)distance; ✓
- Syntax: (Type_Name) Expression
  - Note that the value is truncated, not rounded
  - Note: in the example, the variable *distance* is not changed, the assignment statement affects only the value stored in *points*

Examples of Type Casting

- 3 / 2 = 1
  - Integer division truncates the results
- (double)3 / (double)2 = 1.5
- Try it yourself
  - System.out.println(3/2);
  - System.out.println((float)3 / (float)2);
- What happens if you cast a double into int?
  - E.g.: what's the output of the following statement?

    System.out.println((int)1.5);

Try It Yourself

- Run code in Eclipse
  - See TypeCasting.java on the course website for more details

Arithmetic Operators

- Unary operators
  + : Unary plus operator; indicates positive value
  - : Unary minus operator; negates an expression
  ++ : Increment operator; increments a value by 1
  -- : Decrement operator; decrements a value by 1
  ! : Logical complement operator; inverts the value of a boolean
- Binary arithmetic operators
  *, /, %, +, -

E.g.: rate * rate + delta
1 / (time + 3*mass)
(a – 7) / (t + 9 * v)

% Operator

- Remainder operator, or modulus operator
- The % operator gets the remainder after division
- An example
  - An integer n is even if n%2=0, odd if n%2=1
- Floating-point numbers
  - Java allows to use % with floating-point operands
  - f % d = f – d * q (q is the integer portion of f/d, and the sign of q is the same as the sign of f/d)
  - E.g.: -6.5 % 2.0 = -0.5, 6.5 % -2.0 = 0.5

Specialized Assignment Operators

- Combine an arithmetic operator with the simple assignment operator (=) as a shorthand notation
  - E.g.: amount += 5;

    <--> amount = amount + 5;

    amount *= 25;

    <--> amount = amount * 25;

Parentheses and Precedence (I)

- Expressions inside parentheses
  - Tell the computer which operations to perform first, second, and so forth
  - E.g.:

    (cost + tax) * discount

    cost + (tax * discount)

Parentheses and Precedence (II)

- Precedence rules

Highest Precedence

- First: the unary operators +, -, !, ++, and --
- Second: the binary arithmetic operators *, /, %
- Third: the binary arithmetic operators + and –

Lowest Precedence

Boolean operators: ! → && → ||

E.g.: !true && (false || true) || true

Errors in a Program

- Syntax error: grammatical mistake in your program
- Run-time error: an error that is detected during program execution
- Logic error: a mistake in a program caused by the underlying algorithm

Self-Test Questions

- How do you swap the values of two variables, e.g., Integer, or Floating-point?

Next Class

- Lab 0 & 1
- Bring your laptop and textbook
- To-do before the class
  - Review the slides of lecture 2 on creating objects and accessing objects' methods