CS201 Introduction to Java Programming

Variables and Operators

Chengyu Sun
California State University, Los Angeles

Overview

- Variables
- Operators
- Output using `System.out.print`

Example: Sum100

- Calculate the sum of 1, 2, … , 99, and 100

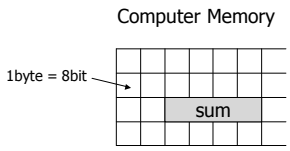
Sum100.java

- Variables
  - sum
  - i

```
public class Sum100 {

  public static void main( String args[] )
  {
    int sum = 0;
    for( int i=1 ; i <= 100 ; ++i )
      sum = sum + i;

    System.out.println( sum );
  }

}
```

Variables

- Name – `sum`
- Type - `int`
- Value - `0`




```
// declaration
int sum;

// assignment
sum = 0;
```

```
// declaration and assignment
int sum=0;

// declare multiple variables
int a, b=1, c;
```

Types

- Class types
- Primitive types
  - `boolean` – true or false
  - `char` – a, b, c, …, A, B, C, …
  - `short` – integers between -2^{15} and $2^{15}-1$
  - `int` – integers between -2^{31} and $2^{31}-1$
  - `float` – single precision real number
  - `double` – double precision real number
- Question: why do we need short when we have int?

Values

- boolean: true, false
- char: 'a', 'b', … , 'A', 'B', …, '1', '2', …
- float, double: -0.1, 99.99, 1.1e13
- short, int: -10, 203, 0x11, 011 …

String

- A sequence of characters enclosed in a pair of double quotes
  - "abcd", "\ " abcd\"", "\\\\" …
- A class type
- In some cases can be used *as if* it is a primitive type

```
String s = "abcd"; // declaration and assignment
String x = "Chengyu";
String y = "Sun";
String xy = x + y; // concatenation
```

More Characters and Strings

- Special character
  - \ – *escape character*
  - '\\' – back slash
  - '\n' – new line
  - '\t' – tab
  - ' ' – white space
- Special (or not so special) strings
  - "" – empty string
  - " " – string with a white space
  - "\"Yes\", he said." – string with double quotes

String and Numerical Types

- Numerical types ➔ String
  - "" + 12 ➔ "12"
  - "" + 123.3 ➔ "123.3"
- String to numerical types
  - Integer.parseInt("12") ➔ 12
  - Double.parseDouble("123.3") ➔ 123.3

Type Mismatch

```
int a = 1000000; // ok
short b = 1000000; // error! overflow

double c = 1000000; // ok. implicit type conversion
int d = 3.6; // error! loss of precision

int e = (int) 3.6; // ok. Type cast
```

- Exercise: write a program which includes the statements above
  - Compile
  - Correct the errors
  - Output the values of a, c, and e

Operators

- Operators
  - =
  - <=
  - ++
  - +

```
int sum = 0;
for( int i=1 ; i <= 100 ; ++i )
  sum = sum + i;
```

Assignment Operator

- =

```
int a; // declaration

a = 10; // assign initial value 10 to a

a = a + 20; // increment a's value by 20
```

Arithmetic Operators

- Addition +
- Subtraction –
- Multiplication *
- Division /
- Reminder %

```
int a = 14;
int b = 4;

int c = a + b;  // ??
int d = a – b;  // ??
int e = a * b;  // ??
int f = a / b;   // ??
int g = a % b;  // ??

int h = a + b – c * d / e % f;
```

Using Parenthesis to Change Evaluation Order

```
int a = 10 + 20 * 2;

int b = (10 + 20) * 2;

int c = (a + b) / 4;
```

Example: Balance Calculation

- User input the initial balance of a CD account
- Annual interest rate 1.2%
- What is the account balance after 3 years?