CSE2016 Programming Methodology

Week 3: Arithmetic and Variables

Instructor: Jinyoung Han (jinyounghan@hanyang.ac.kr)

HANYANG UNIVERSITY

Contents

HANYANG UNIVERSITY

Today's Schedule

1. Integer Arithmetic
2. Named Quantities: Variables
3. Arithmetic with Fractions: Doubles
4. Booleans
5. Operator Precedence
6. Strings, Characters, and their Operations
7. Data-Type Checking
8. Input via Program Arguments
9. Diagnosing Errors
10. Summary

HANYANG UNIVERSITY

Coin Program

- Coins
  - quarter = 25-cent
  - dimes = 10-cent
  - nickels = 5-cent
  - pennies = 1-cent
- An example
  - 9 quarters, 2 dimes, no nickels, and 6 pennies
    - $(9\times25)+(2\times10)+(0\times5)+(6\times1) = 251$ cents = $2.51
- Coin Program

For 9 quarters, 2 dimes, no nickels, and 6 pennies, the total is 251

Coins.java

```
public class Coins
{
    public static void main(String[] args)
    {
        System.out.println("For 9 quarters, 2 dimes, no nickels, and
        6 pennies,");
        System.out.print("the total is ");
        System.out.println( (9 * 25) + (2 * 10) + (0 * 5) + (6 * 1) );
    }
}
```

02. Named Quantities: Variables

HANYANG UNIVERSITY

Variable?

```
public class Coins
{
    public static void main(String[] args)
    {
        System.out.println("For 9 quarters, 2 dimes, no nickels, and
        6 pennies,");
        System.out.print("the total is ");
        System.out.println( (9 * 25) + (2 * 10) + (0 * 5) + (6 * 1) );
    }
}
```

What if # pennies is changed?

We should modify two places!

02. Named Quantities: Variables

HANYANG UNIVERSITY

Variables

- **Place** where values are stored

- Naming rules
  - Alphabet, numbers, _, …
    - E.g., quarters, QUArTers, Q123, my_quarter
  - Do not start with a number
  - Do not use Java keywords: public, class, etc.

- Variable declaration, initialization, and usage
  - int quarters = 5;
  - System.out.println(quarters * 25);

02. Named Quantities: Variables

HANYANG UNIVERSITY

CoinsVariables.java

```
public class CoinsVariables
{
     public static void main(String[] args)
     {
          int quarters = 9;
          int dimes = 2;
          int nickels = 0;
          int pennies = 6;
          System.out.println("For these quantities of coins:");
          System.out.print("Quarters = "); System.out.println(quarters);
          System.out.print("Dimes = "); System.out.println(dimes);
          System.out.print("Nickels = "); System.out.println(nickels);
          System.out.print("Pennies = "); System.out.println(pennies);
          System.out.print("The total is ");
          System.out.println((quarters*25)+(dimes*10)+(nickels*5)+(pennies*1) );
     }
}
```

```
For these quantities of coins:
Quarters = 9
Dimes = 2
Nickels = 0
Pennies = 6
The total is 251
```

02. Named Quantities: Variables

HANYANG UNIVERSITY

Printing Tips

- System.out.print(x); System.out.print(y);

  -> System.out.print(x+y);

- System.out.print(x); System.out.println(y);

  -> System.out.println(x+y);

02. Named Quantities: Variables

HANYANG UNIVERSITY

CoinsVariables2.java

Printing succinctly!

```
public class CoinsVariables2
{
    public static void main(String[] args)
    {
        int quarters = 9;
        int dimes = 2;
        int nickels = 0;
        int pennies = 6;
        System.out.println("For these quantities of coins:");
        System.out.println("Quarters = " + quarters);
        System.out.println("Dimes = " + dimes);
        System.out.println("Nickels = " + nickels);
        System.out.println("Pennies = " + pennies);
        System.out.println("The total is " +
        ((quarters*25)+(dimes*10) +(nickels*5)+(pennies*1)));
    }
}
```

Converting to Dollars

- 251 cents -> $2.51
  - **How to convert cents-scale to dollar-scale?**

- Hint
  - 251/100 = 2 (**quotient**)
  - 251%100 = 51
    - % => **mod**, i.e., remainder operator

- "251" is used twice
  - Let's use a variable, "total"

02. Named Quantities: Variables

HANYANG UNIVERSITY

TotalVariablesDollar.java

```
public class TotalVariablesDollar
{
     public static void main(String[] args)
     {
          int quarters = 9;
          int dimes = 2;
          int nickels = 0;
          int pennies = 6;
          System.out.println("For these quantities of coins:");
          System.out.println("Quarters = " + quarters);
          System.out.println("Dimes = " + dimes);
          System.out.println("Nickels = " + nickels);
          System.out.println("Pennies = " + pennies);
          System.out.print("The total is ");
          int total = (quarters * 25) + (dimes * 10) + (nickels * 5) + (pennies * 1);
          System.out.print("The total is $");
          System.out.print(total / 100);
          System.out.print(".");
          System.out.println(total % 100);
     }
}
```

HANYANG UNIVERSITY

Variable Initialization

- Variable declaration
  - A new “cell” is created
- Variable initialization
  - Initial value is stored in the cell
- We can split “declaration” and “initialization”
  - int dollars;
  - dollars = 3

```
public class TotalVariablesDollar
{
     public static void main(String[] args)
     {
          int quarters = 9;
          int dimes = 2;
          int nickels = 0;
          int pennies = 6;

     }
}
```

02. Named Quantities: Variables

HANYANG UNIVERSITY

Assignment

- Variable value can be changed

```
public static void main(String[] args)
{
      int money = 100;
      System.out.println(money);
      money = 0;
      System.out.println(money);
}
```

```
public static void main(String[] args)
{
      int money = 100;
      System.out.println(money);
      money = money + 50;
      System.out.println(money);
}
```

02. Named Quantities: Variables

HANYANG UNIVERSITY

A Problem: Change Making

- Input: 3 dollars and 46 cents
- Output
  - quarters = 13
  - dimes = 2
  - nickels = 0
  - pennies = 1
- How to calculate?
  - 3.46 – (13 * 0.25) = 0.21
  - 0.21 – (2 * 0.10) = 0.01
  - 0.01 – (0 * 0.05) = 0.01
  - 0.01 – (1 * 0.01) = 0.00

02. Named Quantities: Variables

HANYANG UNIVERSITY

Algorithm

1. Set the starting value of money.
2. Subtract the maximum number of quarters from money, and print the quantity of quarters extracted.
3. Subtract the maximum number of dimes from money, and print the quantity of dimes extracted.
4. Subtract the maximum number of nickels from money, and print the quantity of nickels extracted.
5. The remainder of money is printed as pennies.

HANYANG UNIVERSITY

MakeChange.java

%: modulo

```
public class MakeChange
{
    public static void main(String[] args)
    {
        int dollars = 3;
        int cents = 46;
        int money = (dollars * 100) + cents;
        System.out.println("quarters = " + (money / 25));
        money = money % 25;
        System.out.println("dimes = " + (money / 10));
        money = money % 10;
        System.out.println("nickels = " + (money / 5));
        money = money % 5;
        System.out.println("pennies = " + money);
    }
}
```

03. Arithmetic with Fractions: Doubles

HANYANG UNIVERSITY

Degree Converting

- Degree Converting Problem
  - Celsius into Fahrenheit
  - f = (9/5)c + 32
  - E.g., c = 22 -> f = 71.6

- Fractional numbers representation in Java
  - “double”

03. Arithmetic with Fractions: Doubles

HANYANG UNIVERSITY

CelsiusToFahrenheit0.java

```
public class CelsiusToFahrenheit0
{
    public static void main(String[] args)
    {
        int c = 22; // the degrees Celsisus
        double f = ((9.0/5.0) * c) + 32;
        System.out.println("For Celsius degrees "+ c + ",");
        System.out.println("Degrees Fahrenheit = " + f);
    }
}
```

```
For Celsius degrees 22,
Degrees Fahrenheit = 71.6
```

03. Arithmetic with Fractions: Doubles

HANYANG UNIVERSITY

Type Cast

- Is it ok?

```
int i = 1;
double d = i
```

O

implicit type casting!

- How about this?

```
double d = 1.5;
int i = d;
```

X

```
double d = 1.5;
int i = (int) d
```

explicit type casting!

Booleans

- "True" and "False"
  - boolean b = false;
- Comparison Operations
  - > < <= >= == !=
- Logic Operations
  - "and", "or", "not" -> &&, ||, !
- E.g., x<y && !(y >= 20)

HANYANG UNIVERSITY

Operator Precedence

- 1*2+3
  - (1*2)+3 ? or 1*(2+3) ?

- Priority

unary negation, e.g., -3	High
multiplication, *, division/quotient, /, and modulo %	
addition/string concatenation, +, and subtraction, -	
comparison operations, <, <=, >, >=	
comparisons of equality, ==, and inequality, !=	
logic, &&, ||	Low

05. Operator Precedence

HANYANG UNIVERSITY

Associativity

- 1-2-3
  - (1-2)-3 ? Or 1-(2-3)?
  - We need some operator precedence rules

- General rule
  - Unary operator: right associativity
    - E.g., ----4 = -(-(-(-4)))
  - Binary operator: left associativity
    - 1-2-3 = (1-2)-3

String

- String
  - “Object”
  - String name = “Gildong Hong";

- String concatenation
  - System.out.println("My name is " + name);
  - System.out.println("My name is R2D" + (5-3));

String Methods

- S1.equals(S2)
  - equality comparison – returns whether strings S1 and S2 hold the same sequence of characters
- S1.compareTo(S2)
  - compares the characters in string S1 to S2
- S.length()
  - returns the length of string S
- S.charAt(E)
  - returns the character at position E in S
- S1.indexOf(S2,i)
  - searches S1 for the first occurrence of S2 that appears inside it, starting from index i within S1
- S.substring(E1,E2)
  - returns the substring starting at position E1 and extending to position E2 - 1
- S.toUpperCase(), S.toLowerCase()
  - returns a string that looks like S but in upper-(or lower-)case letters only
- S.trim()
  - returns a string like S but without leading or trailing blanks

- Please refer to the API document!

Characters

- Individual symbols within a string
  - “char”
- Special symbols
  - \b (backspace), \t (tab), \n (newline), \r (return), \" (doublequote), \' (single quote), \\ (backslash)
- Character <-> Number
  - Based on Unicode
  - (char)(‘a’ + 1)?
  - (int)(‘a’ + 1)?

HANYANG UNIVERSITY

Data Type

- Java has two type categories
  - primitive type: int, double, boolean, char
  - reference type or object type: String, GregorianCalendar
- Type error examples

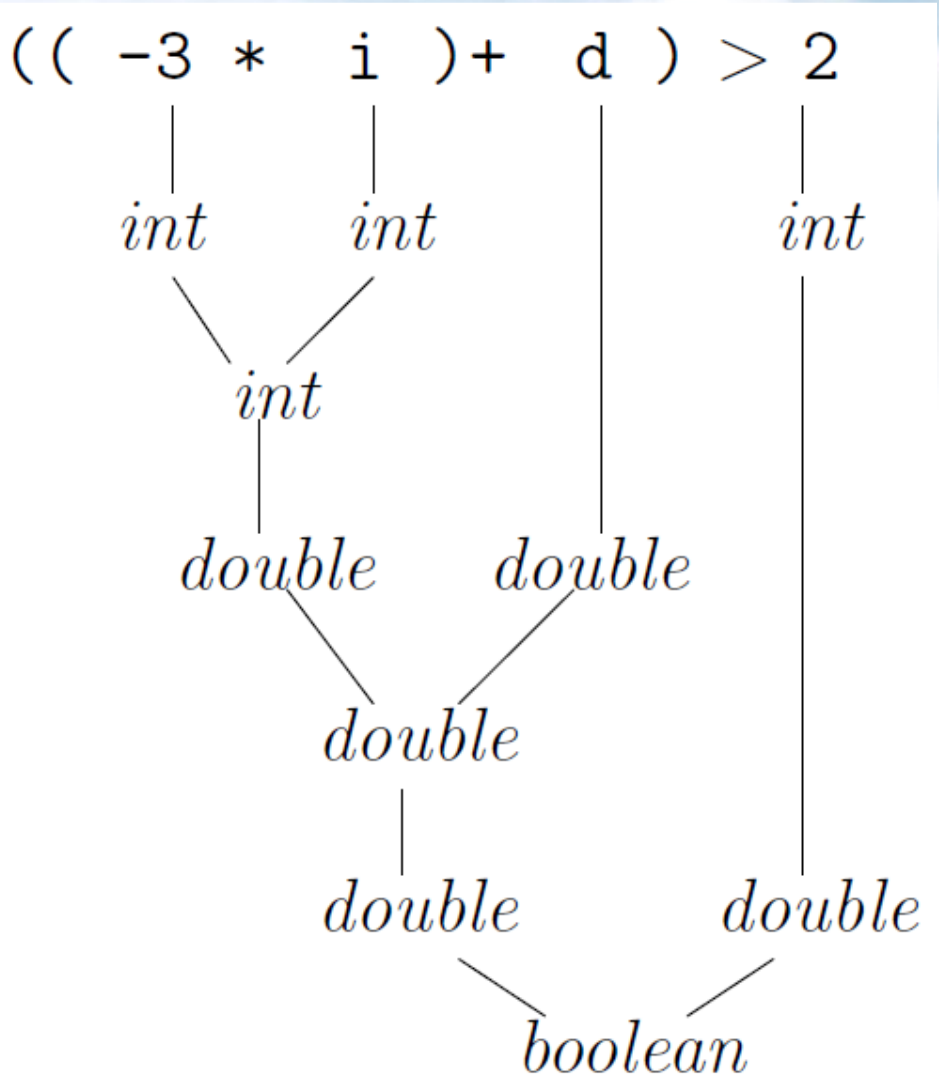
```
boolean b = true;
System.out.println(b * 5); // data type error
int i = 3 * 2.1; // data type error
int x;
x = "abc"; // data type error!

GregorianCalendar c = new GregorianCalendar();
System.out.println(c.getTime());
c.println("oops"); // data type error
```

HANYANG UNIVERSITY

Data Type Tree

int i = 1;
double d = 2.5;



08. Input via Program Arguments

HANYANG UNIVERSITY

Input

- Program arguments (also known as command-line arguments)
- How?
  - Command line: java CelsiusToFahrenheit 20
  - Eclipse: Run configuration > Arguments > Program arguments
- Arguments are stored in args[0], args[1], ..., args[n]

08. Input via Program Arguments

HANYANG UNIVERSITY

LengthOfName.java

```
public class LengthOfName
{
    public static void main(String[] args)
    {
        String name = args[0];
        int length = name.length();
        System.out.println("The name, " + name + ", has length " +
        length);
    }
}
```

08. Input via Program Arguments

HANYANG UNIVERSITY

String <-> Number

- Arguments – **String**
  - E.g., argument 20 -> “20”
- How to change the string “20” into an integer (or double) value?
  - int c = new Integer(“20”).intValue();
  - double c = new Double(“3.14159”).doubleValue();
- Integer, Double ∈ java.lang

08. Input via Program Arguments

HANYANG UNIVERSITY

DecimalFormat

- java.text.DecimalFormat
  - DecimalFormat Object: new DecimalFormat(<pattern>)
  - E.g., pattern = “0.00”

- Using “format” method
  - DecimalFormat f = new DecimalFormat(“0.00”);
  - String s = f.format(100.0/3.0);

08. Input via Program Arguments

HANYANG UNIVERSITY

CelsiusToFahrenheit1.java

```
import java.text.*;

public class CelsiusToFahrenheit1
{
    public static void main(String[] args)
    {
        int c = new Integer(args[0]).intValue();
        double f = ((9.0/5.0)*c) + 32;
        System.out.println("For Celsius degrees " + c + ",");
        DecimalFormat formatter = new DecimalFormat("0.0");
        System.out.println("Degrees Fahrenheit = " +
        formatter.format(f));
    }
}
```

09. Diagnosing Errors

HANYANG UNIVERSITY

Compile-Time Error

- Syntax error
  - Grammar error
  - E.g., System.out.println ((1+2(*3);
  - Compiler can detect it

- Type error
  - E.g., System.out.println(3 + true);
  - Detected by compiler
  - Variable initialization error
    - E.g., System.out.println(a);
    - int a=1; double a=2.5; System.out.println(a);

Run-Time Error

- Errors occurred during run-time
  - E.g., divided by zero error
    - i/x //x=0
  - E.g., conversion error
    - new Integer(x).intValue() // x="abc"

- Java generates an exception for run-time error

```
java.lang.NumberFormatException: abc
    at java.lang.Integer.parseInt(Integer.java)
    at java.lang.Integer.<init>(Integer.java)
    at Test.main(Test.java:4)
```

HANYANG UNIVERSITY

Tricky Error

- Compiler cannot detect the following error
  - int x=3;
  - int y=7;
  - ~~System.out.println(x==y);~~
  - System.out.println(x=y);

10. Summary

Summary

- Variable usage
  - int, double, bool, String, char
- Type
- Input arguments

Thanks

Week 3: Arithmetic and Variables

Instructor: Jinyoung Han (jinyounghan@hanyang.ac.kr)