合肥學院
HEFEI UNIVERSITY

OOP with Java

Homework 01: Expressions

Thomas Weise · 汤卫思

tweise@hfuu.edu.cn · http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Faculty of Computer Science and Technology
Institute of Applied Optimization
230601 Shushan District, Hefei, Anhui, China
Econ. & Tech. Devel. Zone, Jinxiu Dadao 99

合肥学院 南艳湖校区/南2区
计算机科学与技术系
应用优化研究所
中国 安徽省 合肥市 蜀山区 230601
经济技术开发区 锦绣大道99号

Outline

website

Introduction

- We want to practice writing simple Java programs, using typed variables and expressions
- We also want to practice recognizing basic mistakes and pitfalls that might already occur even in this early stage of programming experience
- This homework is comprised of six tasks
- Send me a zip archive named `hw01_[your_student_id].zip` (where `[your_student_id]` is replaced with your student id) with one answer-folder for each homework task (names `hw01-1`, `hw01-2`, ...)

Task hw01-1

- Create a new Eclipse project.
- Inside the project, create a new Java source file.
- In this file, write Java program which declares one local variable of type `long` in its main routine
- Assign value `1000L` to this variable
- Print the value via `System.out.println(...)`
- Assign value `10_00L` to this variable
- Print the value via `System.out.println(...)`
- Assign value `0b1000L` to this variable
- Print the value via `System.out.println(...)`
- Assign value `0x1000L` to this variable
- Print the value via `System.out.println(...)`
- Write a text file with the four printed values and your explanation for the four printed values
- The answer-folder for this task contains the text file and the complete Eclipse project, including source code (.java) and compiled (.class) file.

Task hw01-2

- Marry has 56 golden coins and 5 friends. If she gives the same, maximum (integer) number of coins to each friend, how many coins will she have left?
- Create a new Eclipse project.
- Inside the project, create a new Java source file.
- Into the Java source file, write a program computing and printing the answer.
- The answer-folder for this task contains the complete Eclipse project, including source code (.java) and compiled (.class) file.

Task hw01-3

- We want to write a Java program to compute the base area, surface area, and volume of a cylinder (`https://en.wikipedia.org/wiki/Cylinder`).
- The radius r be 4cm and stored in a variable `r`
- The height h be 3cm and stored in a variable `h`
- First, the base area should be computed, stored in a variable `Ab`, and printed
- Second, the volume should be computed, stored in a variable `V`, and printed
- Third, the surface area should be computed, stored in a variable `As`, and printed
- Do not use Eclipse, write the program with a text editor, compile it with `javac` and run it with `java` in the console/terminal
- The answer-folder for this task contains both the source code (.java) and compiled (.class) file.

Task hw01-4

Listing: What is going on here? (floating point(?) arithmetic)

```
/** A class showing the results of strange (expected) floating point arithmetic expressions. */
public class StrangeFloatingPointArithmetic {
  /** The main routine
   * @param args we ignore this parameter for now */
  public static final void main(String[] args) {
    double a = (10 / 3);                // What result would you expect from 10/3 on a calculator?
    System.out.println(a);              // What did you actually get?
    double b = 1.5d - (0.3d / 0.2d);    // What is 0.3/0.2? What would you expect 1.5 - 0.3/0.2 to be?
    System.out.println(b);              // What did you actually get?
    double c = 1.5d - (3d / 2d);        // OK, same as b, but we use 3/2 instead of 0.3/0.2
    System.out.println(c);              // Do we get the same result for 1.5-(3/2) as for 1.5-(0.3/0.2)?
    boolean d = (0.3d/0.2d) == (3d/2d); // In other words, should this be true or false?
    System.out.println(d);              // What is it actually?
  }
}
```

- Before running the above program, write down your first-glance expectations of the output and why you would expect these values.
- Now run the program. What are the three values printed in the above code?
- For each output value, describe why they are the result of the corresponding expression.
- Did some of them differ from your expectations? If so, what should we do when working with floating point numbers?
- The answer-folder for this task contains the text file with your answers to the above 4 questions.

Task hw01-5

Listing: What is going on here? (integer arithmetic)

```java
/** A class showing the results of strange integer arithmetic expressions. */
public class StrangeIntegerArithmetic {
  /** The main routine
   * @param args we ignore this parameter for now */
  public static final void main(String[] args) {
    int a = 1000;           // declare int a = 1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now 1000*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now (1000*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now ((1000*1000)*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now (((1000*1000)*1000)*1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now ((((1000*1000)*1000)*1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now (((((1000*1000)*1000)*1000)/1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
  }
}
```

- Before running the above program, write down your first-glance expectations of the output and why you would expect these values.
- Now run the program. What are the three values printed in the above code?
- For each output value, describe why they are the result of the corresponding expression.
- Did some of them differ from your expectations? If so, what should we do when working with integer numbers?
- The answer-folder for this task contains the text file with your answers to the above 4 questions.

Task hw01-6

- Why do the following seven programs not compile (with `javac`)? (or if they compile, cannot be executed with `java`)?
- Try to compile each of the programs using `javac`, if they compile, try to run them with `java`
- Store for each program, the compiler error messages (if they compile, the `java` error message) in a separate text file (named like `programName-error.txt`)
- Write one more text file providing – for each of the programs – the reason why they do not compile or cannot be executed.

Task hw01-6: NotCompile1

Listing: Program NotCompile1

```
/** The first program that does not compile: NotCompile1 */
public class NotCompile1 {
  int a = 1000;           // declare int a = 1000
  System.out.println(a); // print value of a, what do we expect?
  a *= 1000;              // a is now 1000*1000
  System.out.println(a); // print value of a, what do we expect?
  a *= 1000;              // a is now (1000*1000)*1000
  System.out.println(a); // print value of a, what do we expect?
  a *= 1000;              // a is now ((1000*1000)*1000)*1000
  System.out.println(a); // print value of a, what do we expect?
  a /= 1000;              // a is now (((1000*1000)*1000)*1000)/1000
  System.out.println(a); // print value of a, what do we expect?
  a /= 1000;              // a is now ((((1000*1000)*1000)*1000)/1000)/1000
  System.out.println(a); // print value of a, what do we expect?
  a /= 1000;              // a is now (((((1000*1000)*1000)*1000)/1000)/1000)/1000
  System.out.println(a); // print value of a, what do we expect?
}
```

Task hw01-6: NotCompile2

Listing: Program NotCompile2

```
/** The second program that does not compile: NotCompile2 */
/** The main routine
 * @param args we ignore this parameter for now */
public static final void main(String[] args) {
  int a = 1000;           // declare int a = 1000
  System.out.println(a); // print value of a, what do we expect?
  a *= 1000;              // a is now 1000*1000
  System.out.println(a); // print value of a, what do we expect?
  a *= 1000;              // a is now (1000*1000)*1000
  System.out.println(a); // print value of a, what do we expect?
  a *= 1000;              // a is now ((1000*1000)*1000)*1000
  System.out.println(a); // print value of a, what do we expect?
  a /= 1000;              // a is now (((1000*1000)*1000)*1000)/1000
  System.out.println(a); // print value of a, what do we expect?
  a /= 1000;              // a is now ((((1000*1000)*1000)*1000)/1000)/1000
  System.out.println(a); // print value of a, what do we expect?
  a /= 1000;              // a is now (((((1000*1000)*1000)*1000)/1000)/1000)/1000
  System.out.println(a); // print value of a, what do we expect?
}
```

Task hw01-6: NotCompile3

Listing: Program NotCompile3

```
/** The third program that does not compile: NotCompile3 */
public class NotCompile3 {
  /** The main routine
   * @param args we ignore this parameter for now */
  public static final void main(String[] args) {
    int a;                    // declare int a = 1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;                // a is now 1000*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;                // a is now (1000*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;                // a is now ((1000*1000)*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;                // a is now (((1000*1000)*1000)*1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;                // a is now ((((1000*1000)*1000)*1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;                // a is now (((((1000*1000)*1000)*1000)/1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
  }
}
```

Task hw01-6: NotCompile4

Listing: Program NotCompile4

```
/** The fourth program that does not compile: NotCompile4 */
public class NotCompile4 {
  /** The main routine
   * @param args we ignore this parameter for now */
  public static final void main(String[] args) {
    int a = 1000;          // declare int a = 1000
    System.out.printf(a); // print value of a, what do we expect?
    a *= 1000;             // a is now 1000*1000
    System.out.printf(a); // print value of a, what do we expect?
    a *= 1000;             // a is now (1000*1000)*1000
    System.out.printf(a); // print value of a, what do we expect?
    a *= 1000;             // a is now ((1000*1000)*1000)*1000
    System.out.printf(a); // print value of a, what do we expect?
    a /= 1000;             // a is now (((1000*1000)*1000)*1000)/1000
    System.out.printf(a); // print value of a, what do we expect?
    a /= 1000;             // a is now ((((1000*1000)*1000)*1000)/1000)/1000
    System.out.printf(a); // print value of a, what do we expect?
    a /= 1000;             // a is now (((((1000*1000)*1000)*1000)/1000)/1000)/1000
    System.out.printf(a); // print value of a, what do we expect?
  }
}
```

Task hw01-6: NotCompile5

Listing: Program NotCompile5

```
/** The fifth program that does not compile: NotCompile5 */
public class NotCompile5 {
  /** The main routine
   * @param args we ignore this parameter for now */
  public static final void main(String[] args) {
    byte a = 1000;          // declare int a = 1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now 1000*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now (1000*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now ((1000*1000)*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now (((1000*1000)*1000)*1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now ((((1000*1000)*1000)*1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now (((((1000*1000)*1000)*1000)/1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
  }
}
```

Task hw01-6: NotCompile6

Listing: Program NotCompile6

```
/** The sixth program that does not compile: NotCompile6 */
public class NotCompile6 {
  /** The main routine
   * @param args we ignore this parameter for now */
  public static final void main(String[] args) {
    int a = 1000;           // declare int a = 1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now 1000*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000d;             // a is now (1000*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now ((1000*1000)*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a /  1000;              // a is now (((1000*1000)*1000)*1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now ((((1000*1000)*1000)*1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now (((((1000*1000)*1000)*1000)/1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
  }
}
```

Task hw01-6: CanCompileButNotRun

Listing: Program NotCompile1

```
/** The seventh program: "CanCompileButNotRun"
    It does compile, but java CanCompileButNotRun will fail */
/** The main routine
 * @param args we ignore this parameter for now */
public class CanCompileButNotRun {
  public static final void main() {
    int a = 1000;           // declare int a = 1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now 1000*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000d;             // a is now (1000*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a *= 1000;              // a is now ((1000*1000)*1000)*1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now (((1000*1000)*1000)*1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now ((((1000*1000)*1000)*1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
    a /= 1000;              // a is now (((((1000*1000)*1000)*1000)/1000)/1000)/1000
    System.out.println(a); // print value of a, what do we expect?
  }
}
```

谢谢

Thank you

Thomas Weise [汤卫思]
tweise@hfuu.edu.cn
http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Institute of Applied Optimization
Shushan District, Hefei, Anhui,
China

Caspar David Friedrich, “Der Wanderer über dem Nebelmeer”, 1818
http://en.wikipedia.org/wiki/Wanderer_above_the_Sea_of_Fog