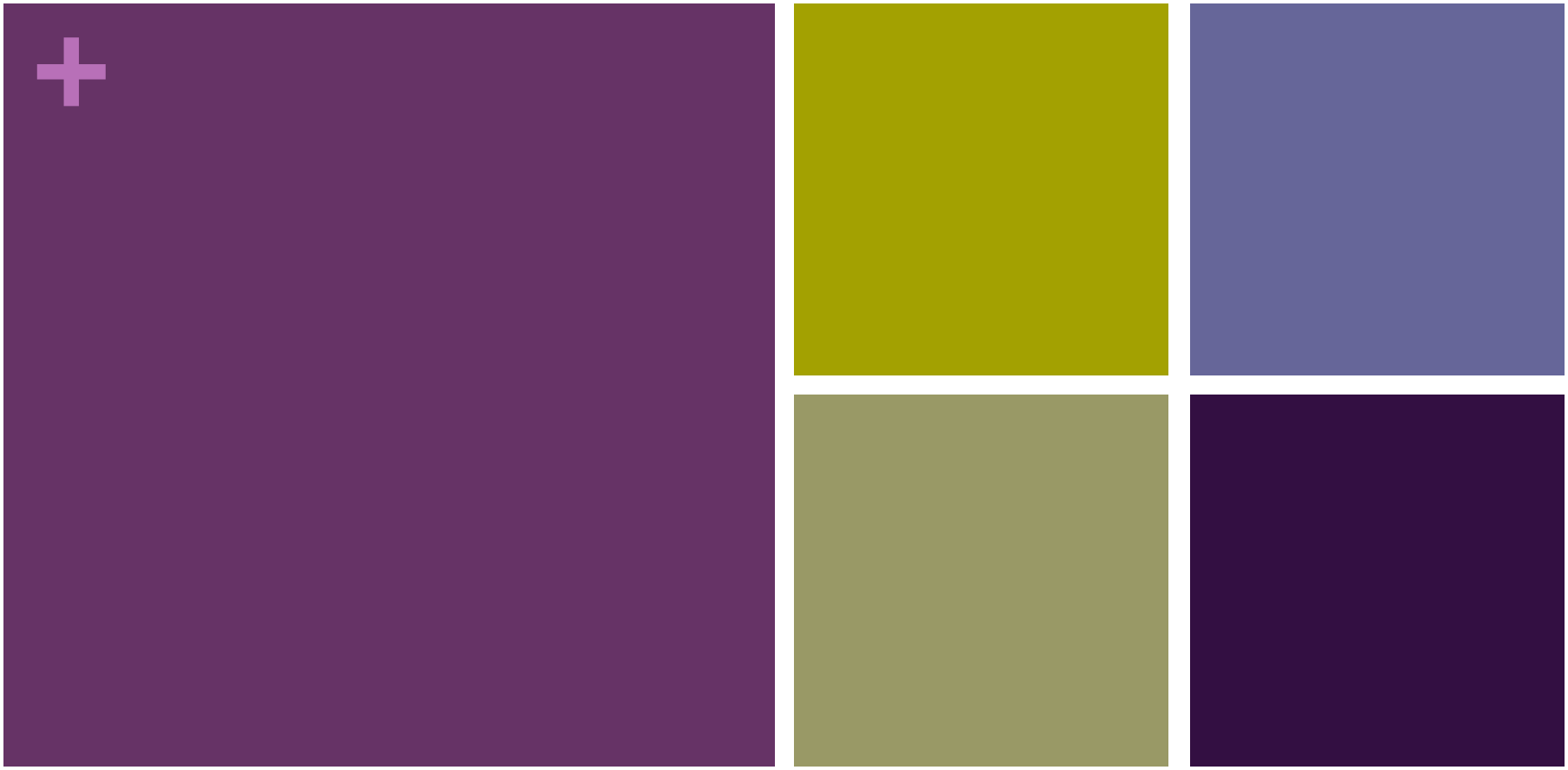


Arithmetic Operators

Deborah A. Trytten

Announcements & Reminders

- Reminder: Canvas can send you message by email or text
  - Account → Settings → Ways to Contact
- Lab 1 tomorrow
  - Project 1 posted early
  - Installing Eclipse!
- Office and tutoring hours coming very soon
  - Will be posted on syllabus and Canvas

Review and Preview

- Last week: storing data
  - What types of data can we store in our programs?

    int, double, boolean, char, String
  - How do we store it? Declare variables and assign values:

    ```
    int ultimateAnswer = 42;
    double gigawatts = 1.21;
    ```

- This week: manipulating numerical data (ints and doubles)
  - Addition, subtraction, multiplication (work as expected)
  - Division (surprising behavior with ints)
  - Modulo operator: % (this may be new to you)
  - Mixed-type operations (2 + 1.5 = ?)
  - Order of operations (1 + 2 / 3 = ?)
  - Type changes (casting)

Mathematical Operators

- Same essential operators in Java as in Math
  - Addition: +
    - Unary: +3
    - Unary: + 7.2491
    - Binary: 3 + 5
    - Binary: 2.7 + 4.1
  - Subtraction: -
    - Unary: -3
    - Unary: -4.298
    - Binary: 6 - 8
    - Binary: 2.7 – 4.1

Multiplication

- Multiplication *
  - Must be explicitly used—no juxtaposition allowed
  - Binary: 5 * 3
  - Binary: 2.2 * 0.5

Critical Difference: Java/Math

- ***int and double are distinct types***
- In Java the values 4 and 4.0 are not the same number
  - 4 is an int
  - 4.0 is a double
- Rules
  - Operate on two int values, get an int result
  - Operate on two double values, get a double result
  - Operate on an int and a double, what then?

Examples

3 + -8

2.1 * 3.0

7 * 4

1.3 + 1.7 // careful

2.4 – 3.9

- What operator haven't we mentioned?

Think, Pair, Share

- Find the values for

6 - 9

12.3 * 10.0

4.2 + 5.8

iClicker Question

- What is 1.2 * 10.0?

Answer a: 12.0

Answer b: 12

Answer c: true

Answer d: “12”

Answer e: ‘12’

Division

- double division works as expected
- 36.0 / 1.2
- int division does not work exactly as expected
  - 10 / 2
  - 11 / 2
- Rule: Integer division truncates any fractional part
  - Source of many program bugs

Examples

39 / 4

42 / 5

9 / 10

99 / 100

9999/10000

9999999/10000000

Think, Pair, Share

- What does each computation produce?

24.0 / 1.2

15 / 3

16 / 3

17 / 3

18 / 3

iClicker Question

- What is 29 / 10 ?

Answer a: 2

Answer b: 3

Answer c: 2.9

Answer d: 2.0

Answer e: 3.0

An Unexpected Operator: %

- The quotient remainder theorem for integers
  - Given any integer A and a positive integer B, there exist unique integers Q and R such that
  - $A = B * Q + R$ where $0 \leq R < B$
- This theorem says that long division works
- Restatement: When you divide A by B, you get a quotient Q and a remainder R
  - The remainder is positive and less than B

Process

- Consider 29 / 3 (show long division)
- 29 = 3 * 9 + 2
  - 9 is the quotient (29 / 3)
  - 2 is the remainder (29 % 3)

Practice / and % Operations

- Work these on your own then compare answers with people in your row until you all agree
  - 5 / 2
  - 5 % 2
  - 37 / 5
  - 37 % 5
  - 93 / 7
  - 93 % 7

iClicker Question

- What is 53 % 9?

Answer a: 5

Answer b: 6

Answer c: 7

Answer d: 8

Answer e: Something else

Is % Useless?

- Suppose it's 11 a.m.
- What time is it in 3 hours?

- % gets used a lot
  - But it always seems surprising
- It's used to generate (pseudo) random numbers
- It's used in cryptography (encoding data)

Assignment Revisited

int size = 3 + 2;

- Expression on right has its value calculated
- This value is stored in the memory location size on the left
- What happens with variables on the right?
  - Use their value

Examples

```
int x = 9;

int y = x + 3;

y = y – 2;

y = y * 2;
```

Show memory diagram

Think, Pair, Share

```
int size = 9;

int count = 2;

int limit = 7;
```

- What is in size, count, and limit at the end of these expressions (executed sequentially in order)?

```
size = limit - count;

count = limit % size;

limit = limit – 5;
```

Announcements & Reminders

- Office hours posted on Canvas
  - Wednesday hours: 11-12, 1:30-2:30
  - Location: Devon Energy Hall (DEH) 115
- Assignments due on Labor Day
  - zyBook chapters 1-4
  - TC 1 and TC 2
  - Project 1
- Homework 1 posted (due Sept 5)
  - Covers last two class topics:
    - Storing primitive data
    - Arithmetic operators

Review and Preview

- Last time:
  - Five arithmetic operators: +, -, *, /, %
  - Operate on two ints → get an int
    Operate on two doubles → get a double
  - Integer division truncates decimal part
    - 7.0 / 4.0 → 1.75
    - 7 / 4 → 1
  - Modulo (i.e., remainder) operator returns remainder
    - 7 % 4 → 3
- Today:
  - Order of operations (1 + 2 / 3 = ?)
  - Mixed-type operations (2 + 1.5 = ?)
  - Type changes (casting)

Write Code

- Suppose we have the following variables:
  - final int BOXES_PER_CRATE = 6; // constant!
  - int boxes = 34;
- Boxes must be shipped in crates, but a crate can only be shipped if it is full.
- Fill in the right-hand side of these statements:
  - int cratesShipped = ?
  - int unshippedBoxes = ?

Precedence

- What is 3 + 2 * 5 in math?
  - (3 + 2) * 5 or 3 + (2 * 5) ?
- Precedence rules for Java

1. ()
2. Unary -/+
3. *, /, % have the same precedence and are done left to right
4. Binary +/ - have the same precedence and are done left to right
5. = (assignment statement)

Examples (similar to Homework 1)

2 / 5 + 3 / 5

5 * 3 % 9 – 4 / -2 + 7

iClicker Question

- Which expression(s) below is the average (mean) of int values x, y, and z? The mean of 3, 5, and 6 is 4.666.

Answer a: x + y + z / 3

Answer b: (x + y + z) / 3

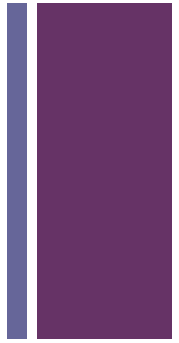
Answer c: x/3 + y/3 + z/3

Answer d: None of the above

Mixing int and double

- When int and double are combined by a mathematical ***OPERATOR***
  - The int is promoted to a double
- This is done at the operator level—one at a time—not at the expression level

Examples



4 / 5 * 7.0

4.0 / 5 * 7

4 / 5.0 * 7

Underlying Principle

- The principle underlying promotion
  - You always know which double is meant by an int
  - You don't know which int was meant by a double
- Java generally lets you do things that are unambiguous

```
double size = 4; // I know you meant 4.0!

int count = 4.3; // Did you mean 4 or 5? Help me!
```

Think, Pair, Share

- What is the result of the computation below?

10 / 3 * 2.0

iClicker Question

- Suppose the variable size is declared with the following statement:

  int size;

Which assignment statement below is legal?

Answer a: size = 3 + 7 – 2.0;

Answer b: size = 2 / 3;

Answer c: size = 10.0 / 2;

Answer d: size = 2 * 5 – 10.0;

Answer e: All of those statements are illegal.

Forcing Type Changes

- You may use a "cast" to force type changes
  - (int)
  - (double)
- Example:
  - int age = (int) 37.3; // Not great code!
- When double is cast to int, the fractional part is truncated
  - Consistent with integer division?
- Casts have precedence greater than multiplicative operators and less than unary + and -

Think, Pair, Share

- iClicker question revisited: write an expression that finds the average (mean) of int values age1, age2, and age3?

- Find (at least) three ways to do this

Observation on Precedence

- Unary operators have higher precedence than binary operators
  - Restatement: Unary operators are sticky
- Multiplicative operators have higher precedence than additive operators
- These rules will generally apply, even as we add operators

Comparison

- Where do Java and Arithmetic work similarly?
- Where do Java and Arithmetic work differently?

- List a few things in each category
- Compare your list with your neighbor
  - Be ready to share

Write a Program

- Write a program in Java that finds the average of 3 project grades
- Strategies:
  - Start small
  - Fix compilation errors first
  - Test often and carefully
    - Figure out test results before running program
  - Once small cases are working, can increase size