

## Lesson 2.01 — Basic Data Concepts

### Overview

Objectives — *Students will be able to...*

- **Identify and categorize** data types
- **Identify** operators and operands.
- **Correctly apply** rules of precedence

Assessments — *Students will...*

- **Write** code that yields a given answer, using rules of precedence
- **Create** expressions and predict output using operator/operand expression sets

Homework — *Students will...*

- **Complete** self-check questions 1-3 (4th edition: 1, 3, 4)
- **Read** BJP 2.2 up to “String Concatenation”

### Materials & Prep

- **Projector and computer**
- **White paper and markers**
- **Classroom sets** of operator/operand expression cards created from WS 2.1
- **Pair or small group** student assignments

Operator/Operand Expression sets can be printed and cut from regular printer paper, or you can write them out on construction paper, creating color-coded sets (recommended to prevent cheating and reinforce memory cues).

### Pacing Guide

Section	Total Time
Bell-work and attendance	5min
Introduction to data types	10min
Think-pair-share activity	5min
Introduction to operators and precedence	10min
Evaluating Expressions activity	20min
Check student study lists	5min

### Procedure

Hook your class today by explaining that they’re going to be able to create a calculator by the end of this week.

**Bell-work and Attendance [5 minutes]**

**Introduction to Data Types [10 minutes]**

Begin with a brief lecture about data types.

- To write a more complicated program like a calculator, we need to familiarize ourselves with the different types of data that Java can work with.
- **Type** (or data type): a name for a category of data values that are all related
  - **Primitive data types:** Store the actual value in the variable
  - \* **Type int** describes all whole numbers, or integers (have students name some examples) -int variables take up 4 Bytes of space in memory (see chart below) -Remember to talk about data overflow and what happens if you put too large a value into an int

- \* **Type double** describes all numbers with decimal points (have students give some examples)
  - double variables take up 8 Bytes of space in memory (see chart below)
  - You can remember if something is a double because there are numbers on both sides of a decimal point (like 2 numbers, double numbers)
- \* **Type boolean** describes logical values—this means true or false. There are no other values in type boolean.

## Pacing Guide

Data Type	Total Size	Range of Values
int	4 Bytes	-2,147,483,648 to 2,147,483, 647
double	8 Bytes	approximately 15 significant decimal digits
boolean	1 bit	true or false

- **Reference data types:** store the location in memory of the object the code is referring to -A String variable would be an example of a reference data type
- An **expression** is a simple value, or a set of operations (an equation) that produces a value.
  - One simple example of an expression is value, like 3.14 or 439.
  - Another example of an expression is  $2 + 5.9$ , because it is an operation that produces a value. (Ask students to point out the int, double, and expression in this example.)
- In the expression  $2 + 5.9$ , the plus sign is called an **operator** because the symbol indicates an operation to be performed on one or more values.
- We refer to the values as **operands**—both int and double are operands.

## Think-Pair-Share Activity [5 minutes]

1. While students are finishing writing down definitions in their notes, write an assortment of data type examples on the board.
2. Have students categorize all of the primitive types on the board during a Think-Pair-Share exercise. Remind students to do scratch work in their notebooks, since it will count towards their classwork grade (this encourages everyone to work during the “think” stage of the activity).
3. Bring the class back to whole group, and call on students to share a category for each data type.

## Introduction to Operators and Precedence [10 minutes]

1. Do a quick review of arithmetic operators. Students should be able to volunteer most of these, but you may have to spend some review on mod, especially if your class is not on grade level for mathematics.
2. Ask students for the operators that represent addition and subtraction.
3. Introduce the special symbols we use for the operators multiplication and division.
  - Division has slightly different rules if you’re working in type int:
    - $12 / 5$  evaluates to 2, because even though the calculator shows us 2.4, int doesn’t let us have a decimal point (what type does?)
    - It’s very important to remember that int always drops the part after the decimal point. So even if you evaluated  $39 / 10$ , your answer would be 3, not 4.
4. Introduce the **mod % operator**, and have students work through a few examples with you to practice.
  - In elementary school we called it a “remainder”

- $1079 \% 34$  evaluates to 25, because you get 31 R 25 (34 goes into 1079 34 times, with 25 left over)
  - If you try to get the answer with your calculator, you won't get 25—you'll need to do long division to get the right answer (or, you can get Java to do it!)
5. If we don't use parentheses in our expressions, Java uses **precedence** to decide which operations go first (students will probably mention PEMDAS), and evaluates left-to-right:
- $13 * 2 + 239 / 10 \% 5 - 2 * 2$ 
    - Start left to right,  $13 * 2$  evaluates to 26
    - $239 / 10$  evaluates to 23 (have students do this one to see if they catch the int)
    - Still moving left-to-right, now  $23 * 5$  evaluates to 3, and  $2 * 2$  evaluates to 4
    - $26 + 3 - 4$  evaluates to 25
6. Quick way to round doubles. We just showed that when doing division we drop the decimals so here is a quick way to round doubles:
- for positive numbers:
    - $(\text{int}) (x + 0.5)$
  - for negative numbers:
    - $(\text{int}) (x - 0.5)$

### Evaluating Expressions Activity [20 minutes]

1. Depending on your class size, have students form pairs or small groups
2. Give each pair or small group a Ziploc bag with a set of operand/operator cards.
3. Students should write out the expressions they create, along with the value they evaluate to, in their notebooks.
4. Once students have finished a set, have them repackage the set and trade with another group (or trade in their set with you).
5. Encourage groups to check each others' answers and help each other if they get stuck.

### Check Student Study Lists [5 minutes]

At the end of class, go over student notebooks.

### College Board Topic Question

After this lesson, students will be able to answer questions from the College Board Unit 1 Topic Questions 1.2: Variables and Data Types

### Accommodation and Differentiation

The curriculum does not officially cover the char type since it is not included in the AP subset. However, if your class is progressing quickly, feel free to introduce char into all future examples, worksheets, and tests.

In ELL classrooms, you should give more examples for each type, and spend more time drilling during the introduction and note-taking segments.

### Teacher Prior CS Knowledge

- The AP CS A exam covers a subset of the Java primitive data types. For a more thorough understanding of the Java data types (byte, short, int, long, float, double, char, and boolean) see [http://www.learnjavaonline.org/en/Hello%2C\\_World%21](http://www.learnjavaonline.org/en/Hello%2C_World%21).
- String is not a primitive data type in Java but is a class. Strings in many ways behaves like a primitive data type, for example you can add two Strings together with the + sign. This is a source of confusion for many beginner Java programmers as the language is inconsistent with its treatment of String.

## Video

- BJP 2–1, *Expressions* [http://media.pearsoncmg.com/aw/aw\\_reges\\_bjp\\_2/videoPlayer.php?id=c2-1](http://media.pearsoncmg.com/aw/aw_reges_bjp_2/videoPlayer.php?id=c2-1)
- CSE 142, *Basic Data Concepts* (40:42–49:59) <https://www.youtube.com/watch?v=KZY0S7wpMAg&start=2442>

## Forum discussion

Lesson 2.01 Basic Data Concepts (TEALS Discourse account required)