CS 149

Professor: Alvin Chao

Compiling a Java Program

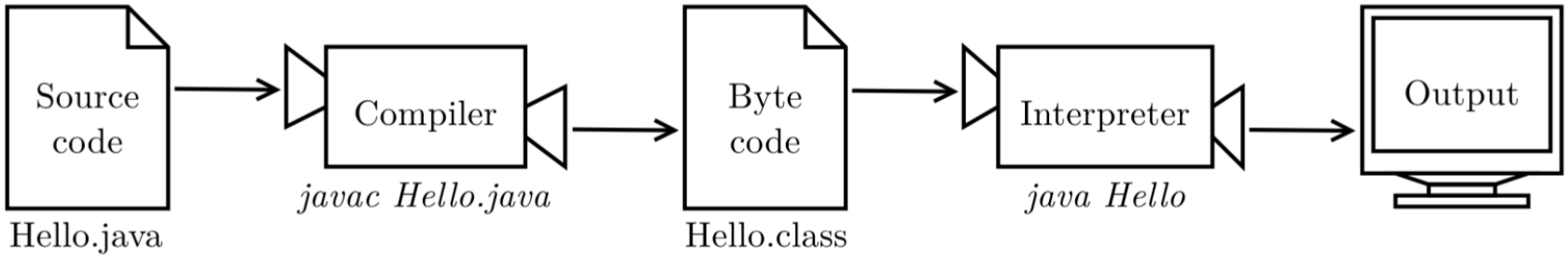

Figure 1.2: The process of compiling and running a Java program.

Java Primitive Types

Java Primitive Types

Keyword	Size	Min Value	Max Value
`byte`	1 byte	-128	127
`short`	2 bytes	-32,768	32,767
`int`	4 bytes	-2^{31}	$2^{31} - 1$
`long`	8 bytes	-2^{63}	$2^{63} - 1$
`float`	4 bytes	$\pm 3.4 \times 10^{-38}$	$\pm 3.4 \times 10^{38}$
`double`	8 bytes	$\pm 1.7 \times 10^{-308}$	$\pm 1.7 \times 10^{308}$
`boolean`	Depends on JVM	`false`	`true`
`char`	2 bytes	`'\u0000'`	`'\uffff'`

Arithmetic Operations Model 1

9 / 4	*evaluates to*	2
10 / 4	*evaluates to*	2
11 / 4	*evaluates to*	2
12 / 4	*evaluates to*	3
13 / 4	*evaluates to*	3
14 / 4	*evaluates to*	3
15 / 4	*evaluates to*	3
16 / 4	*evaluates to*	4

9 / 4.0	*evaluates to*	2.25
10 / 4.	*evaluates to*	2.5
11. / 4	*evaluates to*	2.75
12 / 4.0	*evaluates to*	3.0
13 / 4.	*evaluates to*	3.25
14.0 / 4	*evaluates to*	3.5
15 / 4.0	*evaluates to*	3.75
16 / 4.	*evaluates to*	4.0

1. In the first table, which number(s) divided by 4 evaluate to 3 ? What is significant about the number of answers you have written down?
2. How do the answers in the first table differ from the second table?
3. To the right of the second table, round each answer to the closest integer. How do the values compare to what you see in the first table?
4. Carefully explain the difference between the number formats in the first and second tables.

Model 1 continued

14. / 4.	Evaluates to	
14. / 4	Evaluates to	
14 / 4.	Evaluates to	
14 / 4	Evaluates to	

5. Complete the table
6. Dividing numbers with fractional parts(known as **floating-point** numbers) gives you different results from dividing two integers. In the previous question:
    a) Which rows evaluate to an integer?
    b) Which rows evaluate to a floating-point number?
7. Imagine you are writing a Java program that requires division.
    a) What must be true about the **operands** (the numbers around the **operators**) for you to get the mathematically correct answer?
    b) Does it need to be true for both operands?
8. Consider what you know about addition(+).If you add two integers in a Java expression, will the result always be mathematically correct? Justify your answer.
9. What about subtraction(-) and multiplication(*)? If you subtract or multiply two integers, will the answer always be mathematically correct? Justify your answer.

Model 2

```
1  int dollars;
2  int cents;
3  double grams;
4
5  dollars = 1;
6  cents = 90;
7  grams = 3;
```

10. Identify the Java **keyword** used in a variable declaration to indicate
    a) an integer:
    b) a floating-point number:
11. Consider numbers of dollar bills, cents, and grams. Which of these units only makes sense as an integer, and why?
12. What would you expect the following statements to print out?
    a) System.out.println(dollars);
    b) System.out.println(cents);
    c) System.out.println(grams);
13. What do you think the purpose of a variable declaration is?

Model 2 continued

```
1 int dollars;
2 int cents;
3 double grams;
4
5 dollars = 1;
6 cents = 90;
7 grams = 3;
```

14. Consider the statement: cents = dollars;

a) Compare this code to lines 5–7 in Model 2. What value do you think cents and dollars will have after running this statement?

b) Which side of the equals sign (left or right) was assigned a new value?

15. Examples of Java operators include + and ; they describe an operation to be executed (e.g., addition or subtraction).

a) Do you consider the equals sign in Java an operator (an operation to be executed)? If so, explain the operation. If not, explain why not.

b) Do you consider the equals sign in mathematics an operator (an operation to be executed)? If so, explain the operation. If not, explain why not.

21. In your ownwords, explain how you should read the = sign in Java.For example,theJava statement x = a + b; should be read as “x _____ a plus b.”

Model 3 – Order of Operations

- The Java language defines a specific order of execution for math and other operations. For example, multiplication and division take **precedence** over addition and subtraction. Using parentheses, you can override the order of operations. The following table lists some Java operators from highest precedence to lowest precedence.

Parenthesis	( )
Unary (positive or negative signs)	+ -
Multiplicative	* /
Additive	+ -
Assignment	=

Order of Operations

For the following questions, assume you have these two variables:

int x; double y;

Questions (10 min)

22. What operator has the lowest precedence?Why do you think Java is designed that way?

23. The + and - operators show up twice in the table of operator precedence. For the Java expression x = 5 * -3; explain how you know whether the - operator is being used as an unary or binary operator in this expression.

24. Give the order of operations in the Java expression: a) First operator to be evaluated:

b) Second operator: c) Third operator:

25. Give the order of operations in the Java expression: a) First operator to be evaluated:

b) Second operator:

x = 5 * -3;

26. Based on your answer to the previous question, explain why the variable y would be assigned 4.0 (as opposed to 4 or 4.5).

27. Rewrite the assignment for y so that it would be set correctly to 4.5. (Hint: you'll need to recall what you learned about division in Model 2.)

- **Acknowledgements**
- Parts of this activity are based on materials developed by Helen Hu and Urik Halliday, modified by Chris Mayfield and Nathan Sprague, and licensed under CC BY-NC 4.0 International

---

</end>