

City University of New York (CUNY)

CUNY Academic Works

Open Educational Resources

Kingsborough Community College

2023

Review Java Basics in 2 Weeks (Slides)

Shoshana Marcus

CUNY Kingsborough Community College

[How does access to this work benefit you? Let us know!](#)

More information about this work at: https://academicworks.cuny.edu/kb_oers/37

Discover additional works at: <https://academicworks.cuny.edu>

This work is made publicly available by the City University of New York (CUNY).

Contact: AcademicWorks@cuny.edu

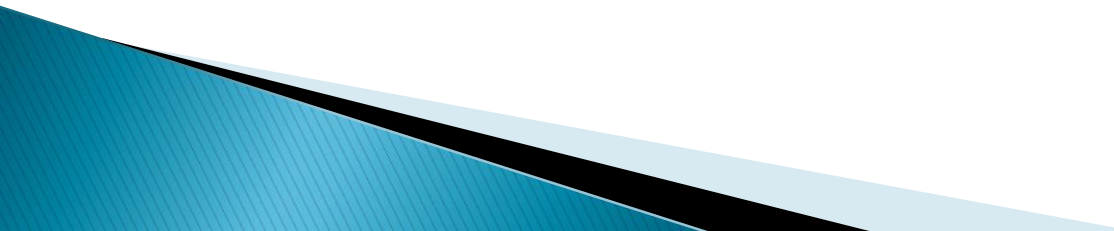
Java Review Part 1

Chapters 1–5

Dr. Shoshana Marcus
Kingsborough Community College


Java programs

Java programs are compiled into bytecode and run by Java Virtual Machine (JVM)

- ▶ Write .java source code
 - ▶ Compile into .class bytecode
 - ▶ Can run same bytecode on any platform that has JVM: *portability* of Java programs.
 - ▶ *What is a compiler?*
- 

Errors

Programming Errors

- ▶ **Syntax errors** – don't speak the language, compiler catches
 - ▶ **Logic errors** – harder to find, write algorithm as pseudocode first
 - ▶ **Runtime errors** – certain scenarios can cause program to malfunction
 - ▶ **TEST your programs with different data as much as you can!**
- 

Java syntax

- ▶ Each statement ends with a ;
- ▶ Java is case-sensitive.
- ▶ Java ignores whitespace but make your programs as readable as possible!

- ▶ Comments

// single line comment

/* multi-line comment */

Java syntax – output

- ▶ Difference between
`System.out.println("Hello");`
and
`System.out.print("Hello");`

- ▶ Difference between
`System.out.println("Hello");`
and
`System.out.println>Hello);`

String literal vs. variable

Variables

- ▶ What is a *variable*? Like a storage box in memory whose contents can keep changing as program runs.
- ▶ To create a variable, specify a data type and a unique name

Example:

```
int numStudents;
```

- ▶ Java is *type-safe* – need to explicitly say what kind of information you have, to make sure you manipulate it in a reasonable manner.

Variables in Java

- ▶ **Variable declaration** is a request to set aside a new variable with a specific name and type. Each variable is declared only once.

```
int numStudents;
```

- ▶ **Assignment statement** stores value in variable. Can perform many times as program runs.

```
numStudents = 17;
```



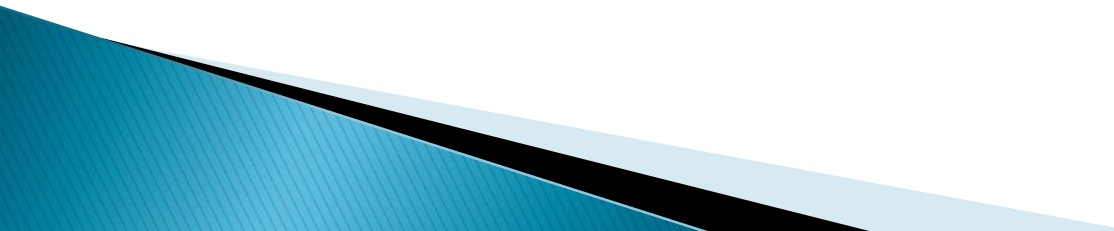
Data types

Primitive data types:

- ▶ boolean
- ▶ char
- ▶ double
- ▶ int

What type of data can be stored in each of these data types?

Naming

- ▶ **Identifiers** name parts of program: class, method, variable
 - ▶ An identifier begins with letter followed by any number of letters or digits or \$, _
 - ▶ Give your variables *meaningful* names!
 - ▶ This ensures that your programs are *self-documenting*.
- 

Expressions

- ▶ Arithmetic operators:

* / + - % ++ --

Combine arithmetic and assignment:

*= /= += -=

- ▶ Division behaves differently on `int` and `double`. Integer division truncates after decimal point. Type of division depends on type of *operands*.

- ▶ Example:

```
double ans = 19 / 2; //9.0
```

```
double ans = 19 / 2.0; //9.5
```

Expressions Explained

- ▶ Arithmetic operators:

* / + - % ++ --

Combine arithmetic and assignment:

*= /= += -=

- ▶ Division by 0 results in runtime error.
- ▶ What does % do? **mod** operator gives remainder after integer division.
- ▶ *What do ++ and -- do?*

Operators

Precedence of operators

* / % precede + -

() force order of operations

Example:

```
int x = 10 + 12 / 5 + 3;    // 15
```

vs.

```
int x = (10 + 12) / (5 + 3); // 2
```

modulus

mod operator: remainder of int division.

Examples:

$$27 \% 20 = 7$$

$$160 \% 20 = 0$$

Exercises:

Write statements that use the % operator

- ▶ What is the remainder of 50 divided by 3?
- ▶ Extract the last digit in a 3-digit number (such as, extract the 7 from 397).

Type casting

Can **cast** a double into an int by truncating after decimal point

Examples:

▶ `(int) 4.75 //becomes 4`

▶ `((int) 2.5) / .5 //4`

Is different than

▶ `(int) (2.5 / .5) //5`

if

- ▶ Conditional execution: sometimes want to execute code but not always
- ▶ test in parentheses determines if body is executed, must evaluate to boolean
- ▶ if body is more than one statement, remember { }

Example:

```
if (hour > 7)
    System.out.println("Wake up!");
```


Boolean

boolean data type: true or false value

Relational operators

- ▶ == != < > <= >=
- ▶ lower level of precedence than arithmetic operators
- ▶ make sure types on each side of operator are compatible
- ▶ evaluate to boolean value

If Exercise

Exercise:

- ▶ Write code that tests if a number is a multiple of 5 and displays “Multiple of 5” if it is.

if / else

Choose between two alternative sets of statements.

- ▶ For multi-line body of if, else, remember { }
- ▶ One condition to test

Example:

```
if(hour > 7)
    System.out.println("Wake up!");
else
    System.out.println("Enjoy your sleep.");
```

Multiple Branching

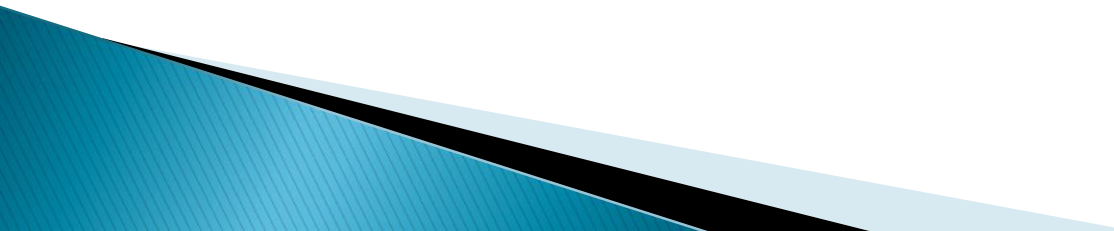
- ▶ `if/ else if /else if /else`
Clearer and more efficient than 3 separate if statements – use whenever possible!

if / else Exercises

Exercises:

- ▶ Write code that increments `x` by 1 if `x` is an odd number, and adds 2 to `x` if it is even.
- ▶ Write code that displays different messages if the variable `num` is a positive number, zero or a negative number.

Boolean logic

- ▶ Nested if /else statements
 - ▶ Indent to make it clear which else and if belong together
 - ▶ Better to use logical AND `&&` and OR `||` than nested if statements.
 - ▶ Review how `&&` and `||` work in Java.
- 

Logical && ||

Come up with values of x and y that satisfy each of these conditions:

- ▶ `if(x > 2 && y > 3)`
- ▶ `if(x > 2 || y > 3)`

If / else

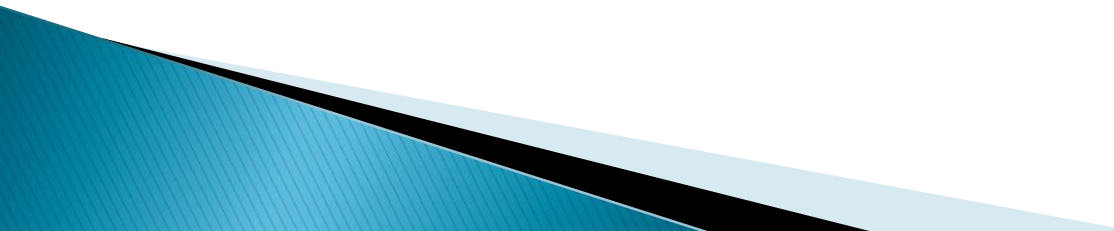
Exercise:

simplify the following segment of code:

```
if ( n > 5 || n == 5 )  
    n = 66;
```


Conditional Execution Ex 1

Exercise:

- ▶ Write a program that sorts 2 integers in ascending order (that the lower number is listed first).
 - ▶ Expand your program to sort 3 integers in ascending order.
 - ▶ Challenge: See if you can sort 3 integers with only 3 comparisons!
- 

Conditional Execution Ex 2

Exercise:

- ▶ Write a program that assigns a letter grade (A, B, C, D, F) based on a student's course average.