Chapter 2

Naming convention → Identifiers ↳ Any word in the program that is not a reserved word.

class name: HelloWorld → Title Case: First letter of each word that makes up class name is a capital letter.

variable name: totalValue

method name: calculateAverage

object name: scn

→ Camel case:

Constant: TAXRATE

final double TAXRATE = 2.9;

keyword
* Declaring a variable as a constant means that you will not be able to change value of this variable.
* This is done by using keyword final in front of the datatype.
* These variable names are all caps, to alert programmer not to inadvertently change value.

Operators

→ Arithmetic: +, -, *, /, %

Left → Right. These are binary Operators. need two operands ↳ 2 + 3

% ↳ modulus

- *, /, % has higher precedence than + & -.

eg: 2/3 + 5 * 6 % 8 / 2 − 3

① ⑤ ② ③ ④ ⑥

Integer division

0 + 5 * 6 % 8 / 2 − 3

0 + 30 % 8 / 2 − 3

6/2 − 3

3 − 3

$\boxed{0}$

5 % 5 = 0
6 % 5 = 1
7 % 5 = 2
8 % 5 = 3
9 % 5 = 4
10 % 5 = 0
11 % 5 = 1
0 % 5 = 0
1 % 5 = 1

Z % n → [0, 1, .. n−1]

Z % 5 → [0, 1, 2, 3, 4]

$$\begin{array}{r} 0 \\ 5\overline{)\,1} \\ 0 \\ \hline R=1 \end{array}$$

→ Relational: These boolean operators setup relationship between two expressions.

==
<
>
<=
>=
!=

= → This is an assignment operator. Has least precedence.

→ ~~≤ or =<~~ → Wrong

!= → not equal to.

eg: boolean tic = 2==3; (valid) / not valid ?

2==3 → false

Logical Operators

AND → (&&)
OR → (||) ESC + \
NOT → (!)

Used to combine more than one boolean statement.
eg: Grade of "B".

g >= 80 && g < 90 ✓

Truth tables – lists all options when you combine 2 or more statements.

A	B	A and B	A || B	!A	!B	!A && !B	!(A || B)
T	T	T	T	F	F	F	F
T	F	F	T	F	T	F	F
F	T	F	T	T	F	F	F
F	F	F	F	T	T	T	T

! (A && B) = !A || !B
! (A || B) = !A && !B

→ De Morgan's Law

H.W

Short Answer
Exercises 2.1 – 2.7
Programs 2.1 – 2.7

Scanner Class

In order to use any class inside another class you need to do the following:

1. Import class
2. Create an object of the class (instantiate)
3. Access methods using the object created in step 2.
   ↳ In order to do this you need to know name, function and parameters of the method.

eg Scanner

1. import java.util.Scanner; → This statement goes outside the class.

Following goes inside the main method

2. Scanner scn = new Scanner(System.in);

- Scanner: is the imported class
- scn: Is an identifier that represents an object of type Scanner
- new: is a keyword, used to create an object of any class.
- Scanner(System.in); Is a method, with the same name as the class.

Notice: Name of this method starts with upper case. (This is an exception)

How do I know that this is a method? – name of the identifier is followed by parenthesis.

Method with the same name as the class is referred to as the "Constructor". This special method is responsible for giving initial values and creating the object of type class.

System.in: Is the input required to create an object of type Scanner

This whole process is reffered to as,
- creating an instance of a class.
- here scn is an instance of type Scanner.

#3 **Using Scanner**
- This is done through instance of Scanner i.e. (scn)
- I also need to know what functions are inside the Scanner class
- I will need to know signature of function as well.

List of all functions provided by the Scanner Class can be found online.
→ Search: Java Scanner "API".

This becomes my handle through which I call various functions of the Scanner class.

Here we will discuss some of the functions of Scanner.

(i) int nextInt()

int → Type of information this function will return.
nextInt() → name of the function.

What does nextInt() do?
⊙ nextInt() is a method of the scanner class and can read integer input from the keyboard.

How do I use this function?
(a) Prompt user to input integer value
(b) Read this value and bring it into the program.

```
System.out.println("Please enter your age"
                   + "as an integer value");
```

```
int age = scn.nextInt();
```

scn an object of type Scanner read next 32 bits of information from the keyboard as an integer.

next page

Scanner methods/function

(ii) String nextLine() → used to read information of type String.

How to use?

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);

    System.out.print("Please enter your name: ");
    String name = scn.nextLine();
```

(iii) double nextDouble();

used to read information of type double from the keyboard.

(iv) void close()

Object of type Scanner has access to "System.in"; i.e. to any input from the keyboard. This can become a security risk if you do not close Scanner object.

After using an object of type Scanner, you must close it.

```
scn.close();
```

Signature of a function

String nextLine()

1. name of the function
2. Type of information returned by the fn
3. Total number, and type and order of parameters if any

In this case

nextLine() does not accept any parameter and will return information of type String

Math class

This class includes all of the math functions that you have studied so far.

Math is a static class. → This means that you do not need to create an object of type Math in order to use this class.

Is an exception. 99.9% classes are not static.

Means that it gets memory as soon as it is declared.

→ Implies that we need to allocate memory and processing resources all the time for this class.

That implies Math is a resource heavy class, so I should not create a class static unless it is absolutely necessary.

Q. So how do we use Math class, since we can not create an object of type Math.

Ans: import

import java.lang.Math;

// Since you can not create an object of type Math, this class itself acts as an handle to
// call various methods.

List of methods inside the Math class.

1. double abs (double a)
2. double pow (double a, double b)

return type; name of method; base; exponent

double val = Math.pow(3,2);

$\boxed{9.0}$

Notice Math is the handle

3. double ceil (double a)

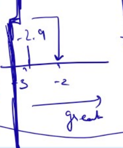
return information of type double; is name of the method; accepts one parameter of type double

In order to use a method, you must know what does that method do.

So what does ceil do? → It round up a value to the closest integer and stores it as a double.

double val = Math.ceil(2.4);

3.0		2.7
3.0		2.9
2		2
-2.0		-2.1
-2.0		-2.9



4. double floor (double a)

double d = floor(2.9);

2.0		2.9
2.0		2.1
-3		-2.1
-3		-2.9

→ rounds down.

5 Constants in the Math class

Math.PI

will give you the value of PI

Math.E

6 long round (double a)

int n = Math.round(2.4)

2		2.4
3		2.5
2		2.19

This is the only method of the Math class that will return an int value. All other methods return double value.

Random Class

import java.util.Random;

Random rand = new Random();

Some methods

- int nextInt(int)

7 double random()

This method returns a real number between 0 and 1. Note 1 is not inclusive

range of values returned [0, 1)

double m = Math.random();

$\boxed{0.113}$

Q: Use Math.random() to generate a random integer between [1, 6] inclusive.

0.1	x6	0.6	(int)	0	+1	1
0.2	x6	1.2	(int)	1	+1	2
0.3	x6	1.8	(int)	1	+1	2
0.4	x6	2.4	(int)	2	+1	3
0.5	x6	3.0	(int)	3	+1	4
0.6	x6	3.6	(int)	3	+1	4
0.7	x6	4.2	(int)	4	+1	5
0.8	x6	4.8	(int)	4	+1	5
0.9	x6	5.4	(int)	5	+1	6
0.99	x6	5.8	(int)	5	+1	6

int x = (int) Math.random()*6 + 1 ;

[0-1)

[0, 6)

[0, 5]

[1, 6]