THE UNIVERSITY OF MELBOURNE

Workshop wk3

COMP90041 Programming and software development

Zhe(Zoe) Wang

Before the workshop

Email: zoe.wang1@unimelb.edu.au

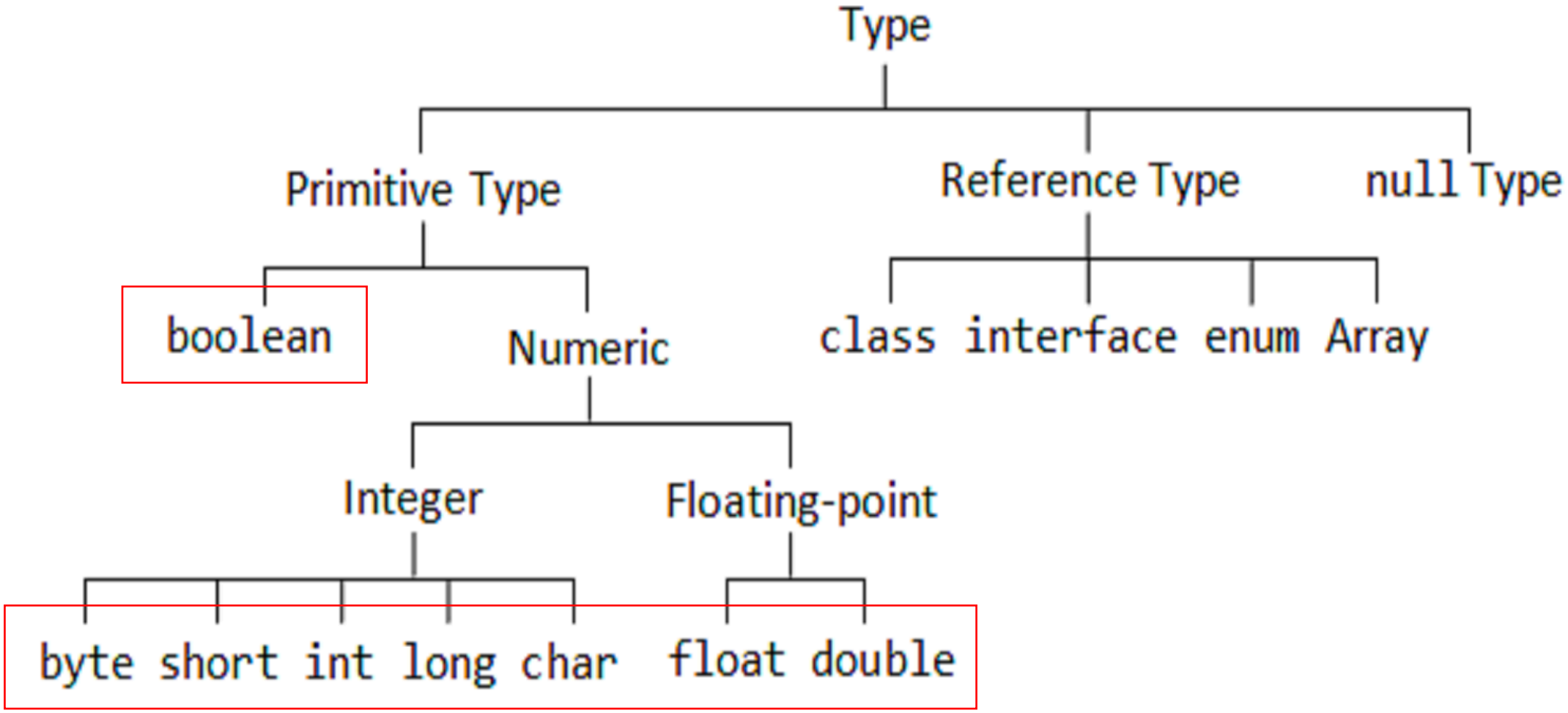
Day	Start Time	Duration	Location
Wed	2:15PM	1h	Zoom
Thur	3:15PM	1h	Zoom
Fri	2:15PM	1h	Zoom

Slides and Demo code can be found in:

https://github.com/Zoeewang/COMP90041-sem1-2021

- download Lab_week3 on canvas

Primitive Types



Primitive Types

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

Operations - for numbers

+ -* / % < <= > >= == !=

%: remainder operator

eg: 10 % 3 = 1

== is a comparison while = is an assignment

```
if (x == 0)?
int x = 0;
```

Operations - for String

\n = newline (end current line), \r – return (go to start of line), \b = backspace, \t = tab character

- Include double-quote in a string by preceding with backslash (\ )

- Include a backslash in a string by preceding with backslash

demo1 print I think "\" is funny

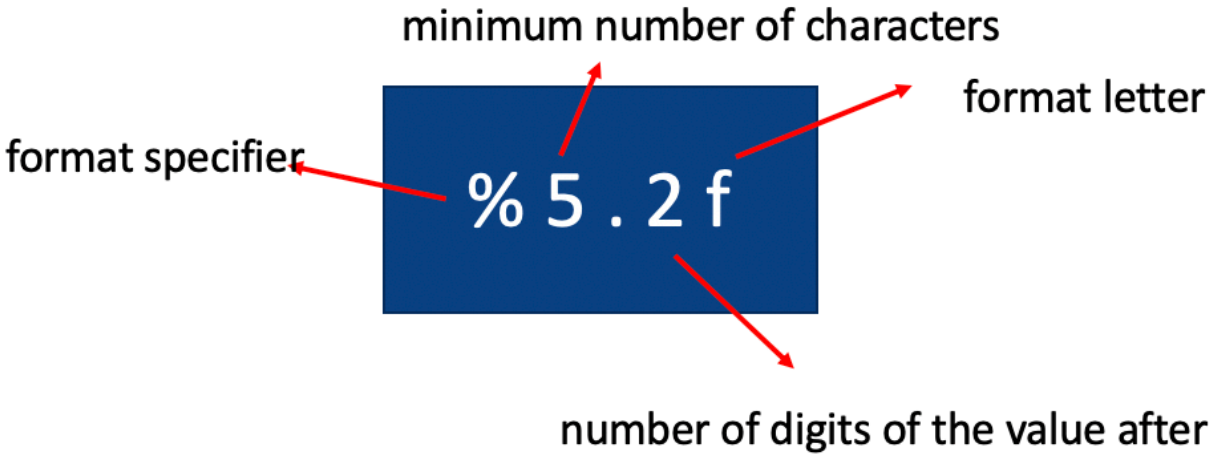
printf – formatted output

- Form:
  `System.out.printf(format-string, args...);`
- *E.g.*:
  `System.out.printf("Average:  %5.2f", average);`

  - If the number if negative, the value will be left-justified, otherwise right-justified

demo2

x =5.7889
print as
5.78



printf – formatted output

d : decimal integer [byte, short, int, long]

f : floating-point number [float, double]

c : character Capital C will uppercase the letter

s : String Capital S will uppercase all the letters in the string

n : newline Platform specific newline character- use %n instead of \n for greater compatibility

Reading console input

```
import java.util.Scanner;
```

```
Scanner keyboard = new Scanner(System.in);
```

```
String line = keyboard.nextLine();
```

What	Type	Expression
One word	`String`	`keyboard.next()`
One integer	`int`	`keyboard.nextInt()`
One double	`double`	`keyboard.nextDouble()`

- After `next`, `nextInt`, or `nextDouble`, `nextLine` just reads rest of current line (maybe nothing!)

demo3

read:

2021
hello
world

Lab 3 Q1

1.Write a program that reads two floating point numbers and print their sum, difference, and product.

eg: input : 0.1 0.2

“0.1 + 0.2 = 0.3”

“0.1 - 0.2 = -0.1”

“0.1 * 0.2 = 0.02”

What is type Casting?

Casting

java can convert types for you automatically

widening conversion（ automatically） converts a number to a wider type

byte -> short -> char -> int -> long -> float -> double

Narrowing Casting (manually) - converting a larger type to a smaller size type

double -> float -> long -> int -> char -> short -> byte

demo4

Math Class

Math.abs(x): Returns the absolute value of x

Math.PI

Math.pow(x, y): Returns the value of x to the power of y

Math.floor(x): Returns the value of x rounded down to its nearest integer

Math.max(x,y): Returns the number with the highest value

demo

Lab 3 Q2

2.Write a program that reads the radius of a sphere and prints its volume and surface area. Use the following formulas, where r represents the radius:

(a) Volume $= \frac{4}{3}\pi r^3$

(b) Surface Area $= 4\pi r^2$

eg: input : 2

“volume : 33.51”

“surface area: 50.27”

handling command line inputs

- When your program is run, it can be given arguments on the command line
- For the boilerplate we've been using, the command line arguments can be referred to as:
  - first command line argument: `args[0]`
  - second command line argument: `args[1]`
  - third command line argument: `args[2]`, *etc.*.
- Each of these is a `String`
- To converts string to int:
  `Integer.parseInt(string)`

demo5

print "hello" + args[0] where args are names

Calculate args[0]+1

if /else

- Form:

  `if (expr) Statement1 else Statement2`

- Executes $Statement_1$ if the `expr` is `true`, else executes $Statement_2$

demo6

- Java also has an if-else expression:

  $expr_1$? $expr_2$: $expr_3$

  - If $expr_1$ is `true` value is $expr_2$
  - If $expr_1$ is `false` value is $expr_3$

Print hello when x is greater than 0 else Print hi

For compound statements:

- Form: { $Statement_1$; ··· $Statement_n$; }
- Don't follow the brace with semicolon

Lab 3 Q3

Write a program that calculates the total wages based on the number of hours worked.

The wages are calculated at a rate of 8.25 per hour for hours less than 40 and at the rate of 1.5 the standard rate for any hours greater than 40.

Number of hours is a command line argument to the program.

if t<40 wage = t * 8.25

if t>40 wage = 40*8.25 + (t-40) *8.25*1.5

Thank you