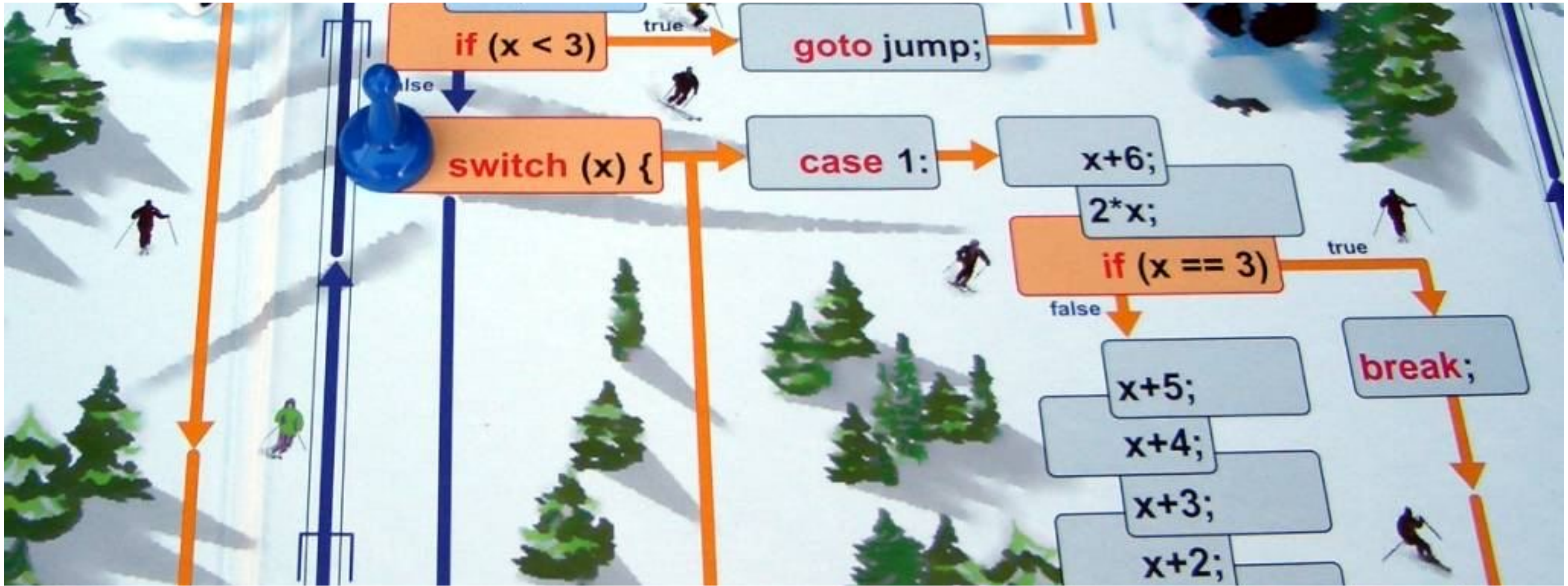


Creative Coding 2023

Instructor: Neng-Hao (Jones) Yu

Course website: https://openprocessing.org/class/83620

Data types (資料型態)

123

3.1415926

456

“Hello World”

“Say Hi”

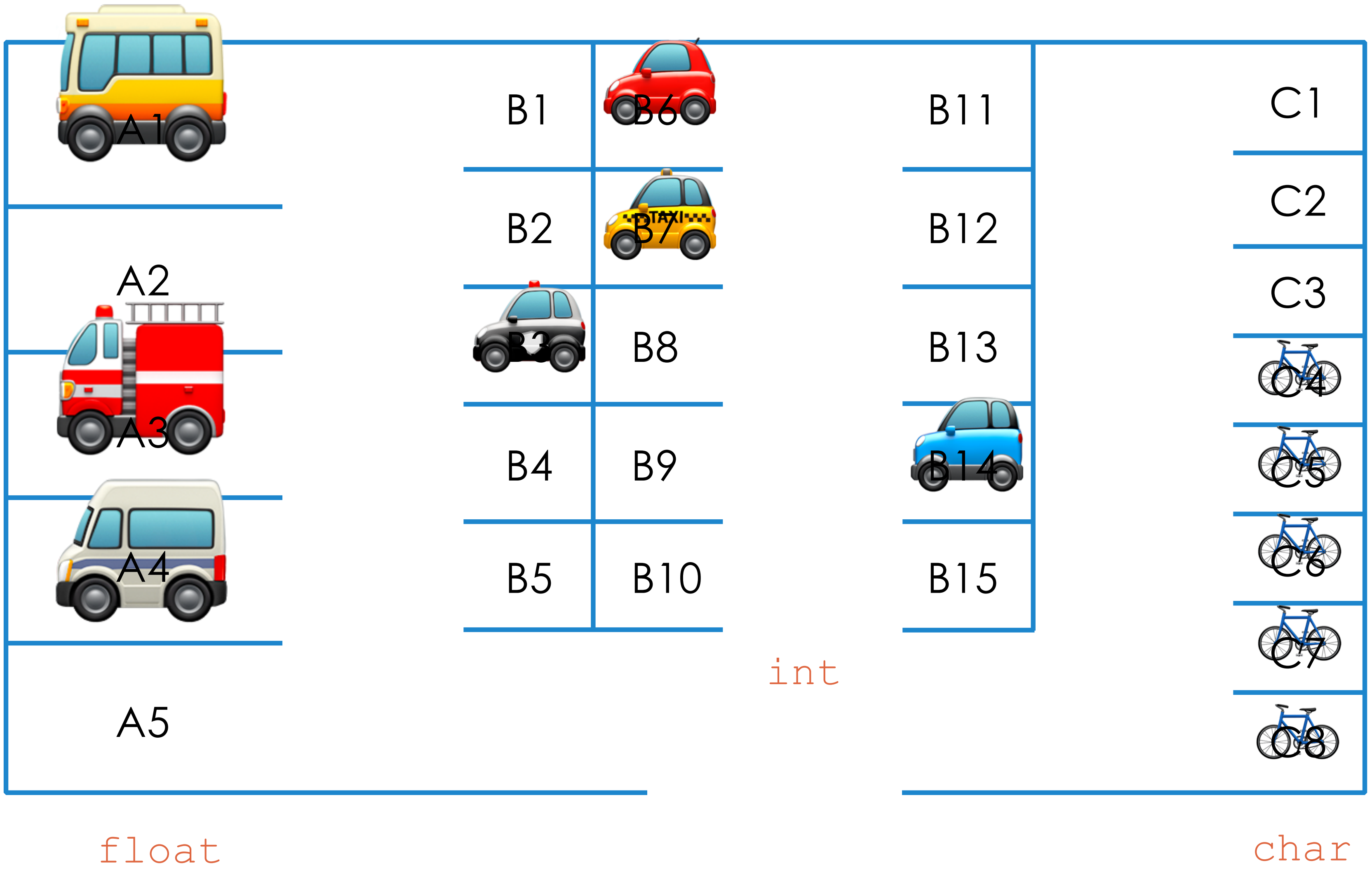
‘c’

Data types

- Different types of data require different types of variables to store them

- The Data Type in Memory
  - `int`: integer
  - `float`: floating-point number
  - `boolean`: boolean variable (true or false)
  - `char`: single character, e.g., ’c’
  - `String`: “string of words”

Memory vs Parking lots



A1
A2
A3
A4
A5

B1 B6
B2 B7
B3 B8
B4 B9
B5 B10

B11
B12
B13
B14
B15

C1
C2
C3
C4
C5
C6
C7
C8

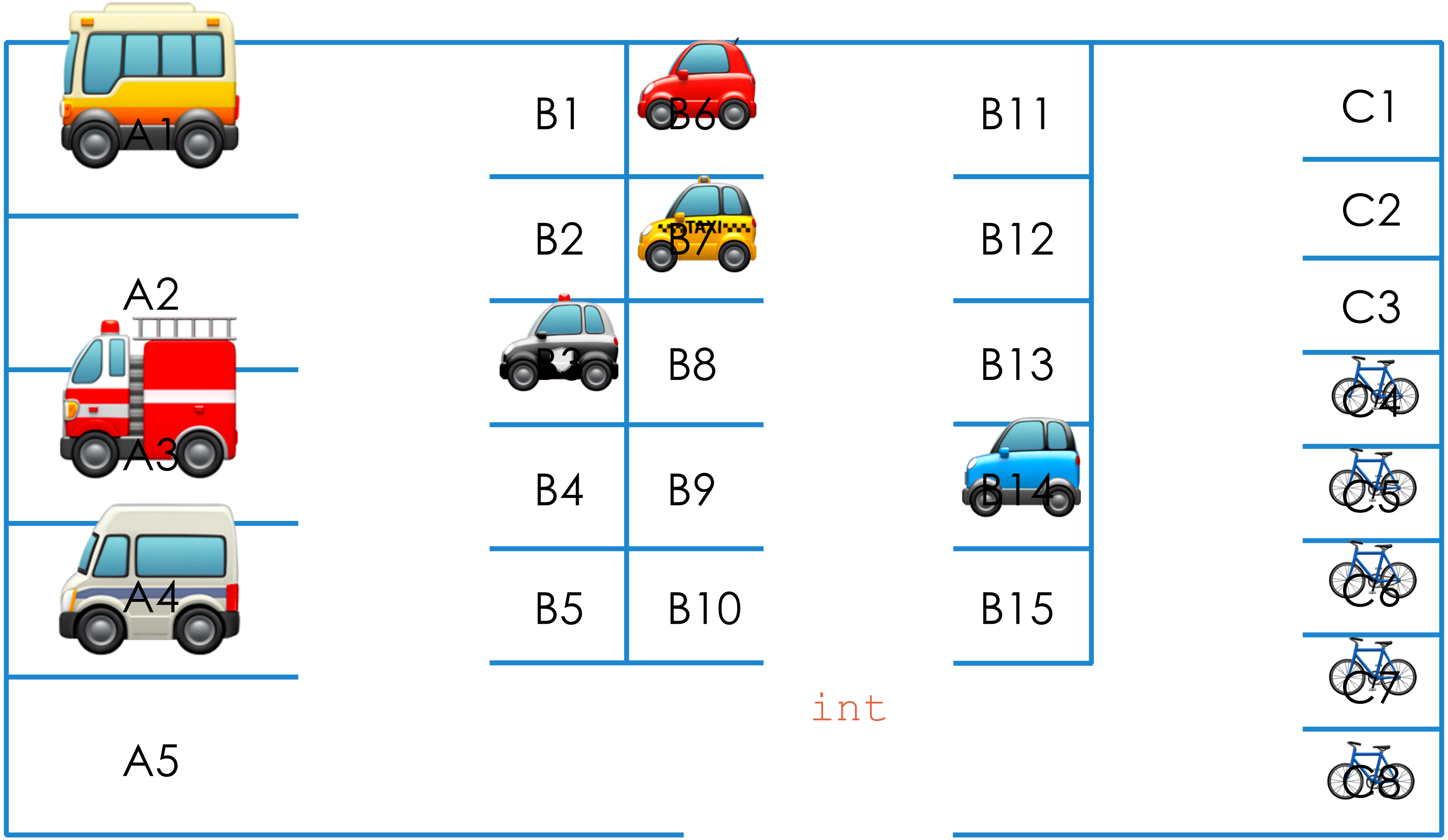
int

float

char

Is it possible to put data of one type into a variable of another type?

Memory and Storage Space



A1

A2

A3

A4

A5

B1 B6

B2 B7

B3 B8

B4 B9

B5 B10

B11

B12

B13

B14

B15

C1

C2

C3

C4

C5

C6

C7

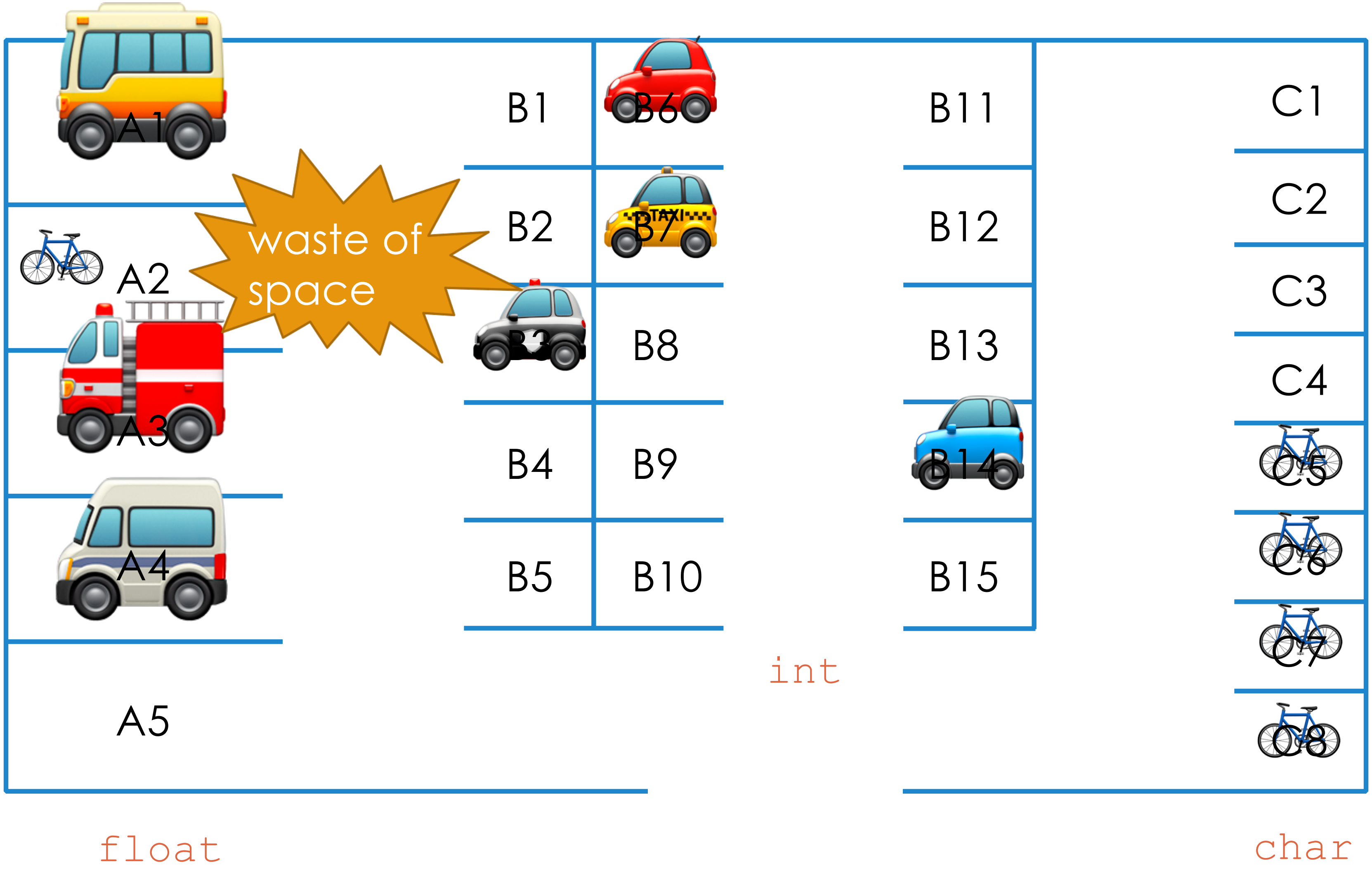
C8

int

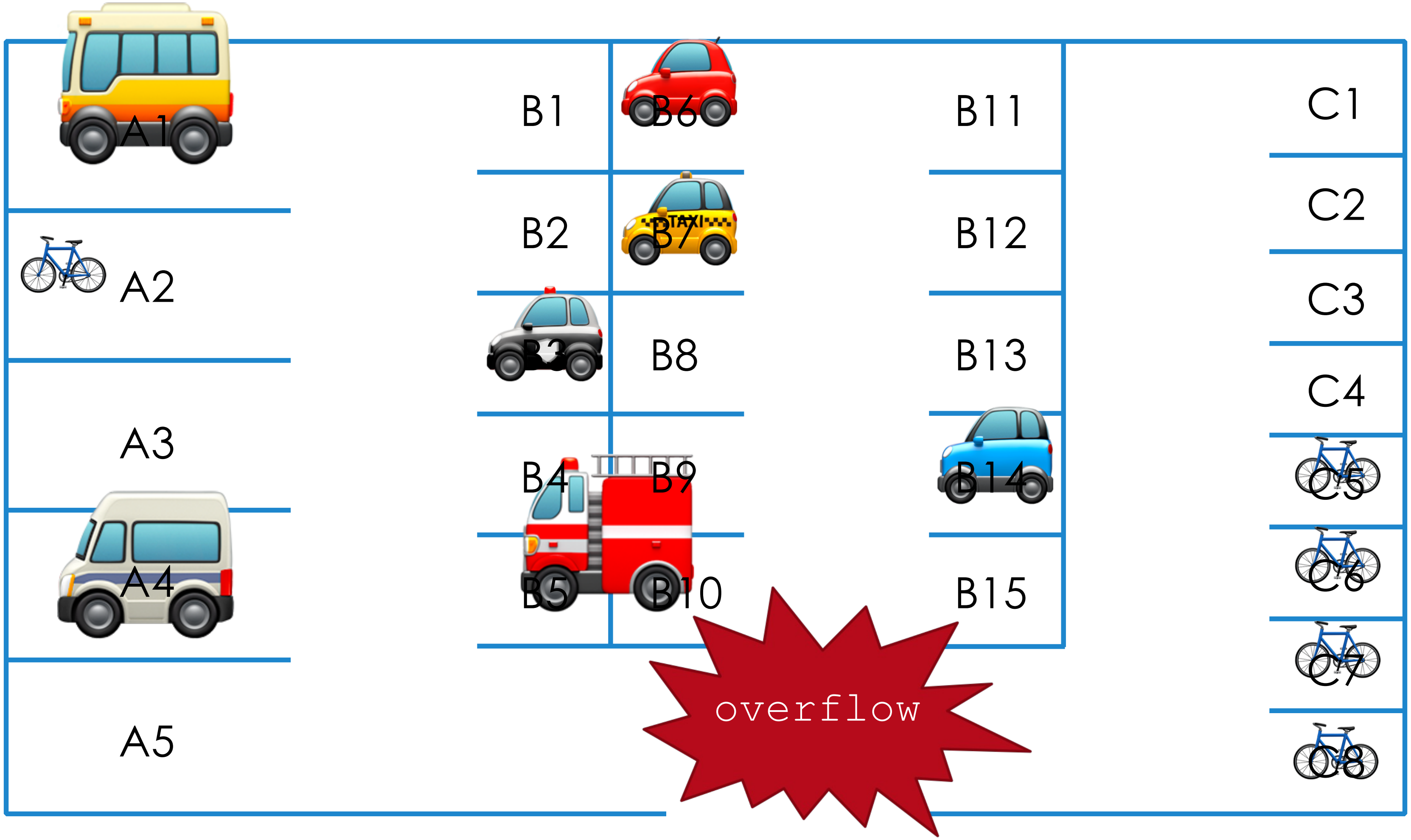
float

char

Memory and Storage Space



Memory and Storage Space

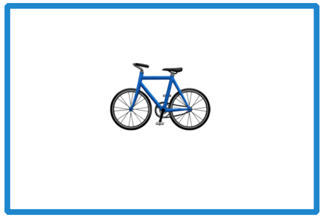


float

char

1. Preemptively checking for potential errors

2. Efficiently planning for memory usage



Advantages of a strongly typed language

Data types

- Size and value range of data types:

overflow
(out of range)

ype	Size	Range of values
t	*32bit*	*-2^{31} ~ $(2^{31}-1)$*
float	*32bit*	*±3.40282347E+38*
char	*8bit*	*A-z, 0-9, and symbols*
boolean	*1bit*	*True / False*
String	*string of words*	*Any word or sentence*
PImage	*PNG or JPG image*	*N/A*

https://zh.wikipedia.org/zh-tw/計算機編碼

Type conversions (型別轉換)

- Implicit conversion, also called **coercion**, is sometimes performed at runtime. It happens in
  - In assignment statements
  - In expressions using certain operators, such as the addition (+) operator

- Explicit conversion, also called **casting**, occurs when your code instructs the compiler to treat a variable of one data type as if it belongs to a different data type.

Question

integer

- 3 / 2 = 1

float

- 3.0 / 2.0 = 1.5
- 3.0 / 2 = 1.5 // Implicit conversion
- 3 / 2.0 = 1.5 // Implicit conversion

```
int a;

float b;

a= 55;

b= a / 2;

println(b);
```

```
//implicit conversion
b= a / 2.0;

//explicit conversion
b= (float)a / 2;

//explicit conversion
b= float(a) / 2;
```

```
// what is the value of b?
27.0
```

operator overloading(運算子多載)

Concatenating Two Strings

```
println("Hello"+"World"); // HelloWorld

println(5 + 5); // 10

println("5" + 5); // “55”
```

Constants (常數)

- Storing Fixed, Unchanging values
- After initialization, constant cannot be changed
- Declaration Method: Add **`final`** before the `datatype`
- Naming Convention: Uppercase Letters with Underscores as Separators

```
final float MILES_KM_CONVERSION_VALUE = 1.61;
final float PI = 3.14;
final int NBA_FOUL_LIMIT = 6;
final int NBA_QUARTER_TIME_LENGTH = 12;
final int NBA_SHOT_CLOCK = 24;
final boolean DEBUG = true;
```

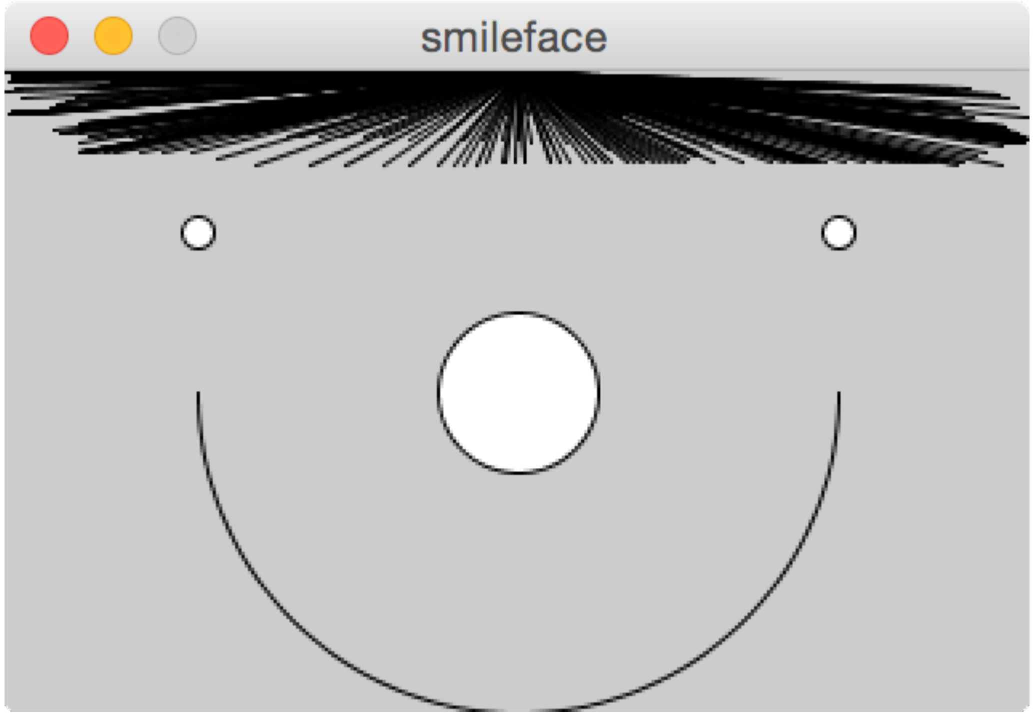
```
// convert mile to km
final float MILES_KM_CONVERSION_VALUE = 1.61;

float km = 30 * MILES_KM_CONVERSION_VALUE;

println(km);
```

Exercise (5mins):
Write a program to calculate the area of a circle given a radius

Build-in constants and variables

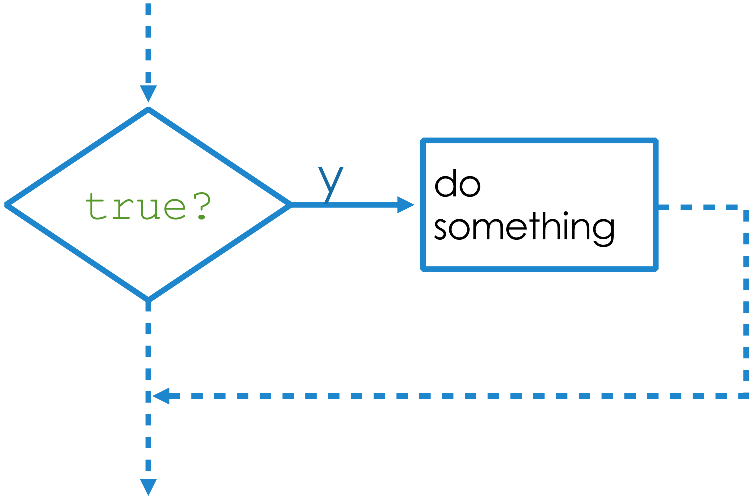
name	*description*
`PI`	*π*
`QUARTER_PI`	*1/4 π*
`width`	*width of the canvas*
`height`	*height of the canvas*
`mouseX`	*current x position of the mouse cursor*
`mouseY`	*current y position of the mouse cursor*
`mousePressed`	*whether or not a mouse button is being pressed*

Exercise



Flow control: conditional statements

```
if (boolean_expression){
    // do something...

}
```

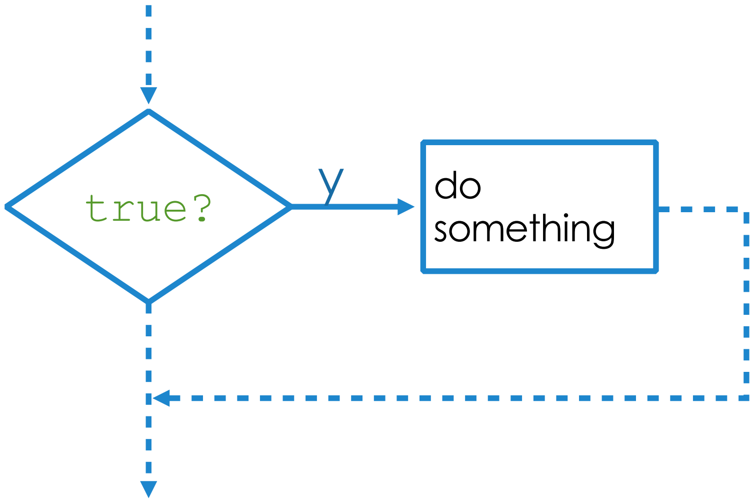


- ❑ Use boolean operators to perform logical operations on boolean values

- ❑ a && b : AND
- ❑ a || b : OR
- ❑ !a : NOT (true ←→ false)

comparison operators

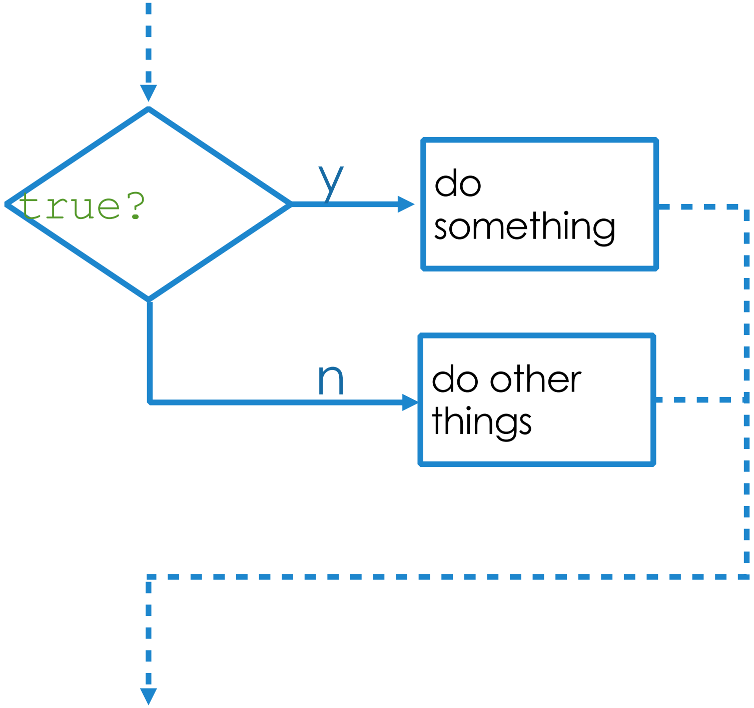
```
if (boolean_expression){
    // do something...

}
```



- ❑ comparison operators
  - ❑ a>b
  - ❑ a<b
  - ❑ a<=b
  - ❑ a>=b
  - ❑ a==b
  - ❑ a!=b

If-else statement & a shorthand way

```
if (boolean_expression){
    // do something...

}


else {

        // do other things...

}
```



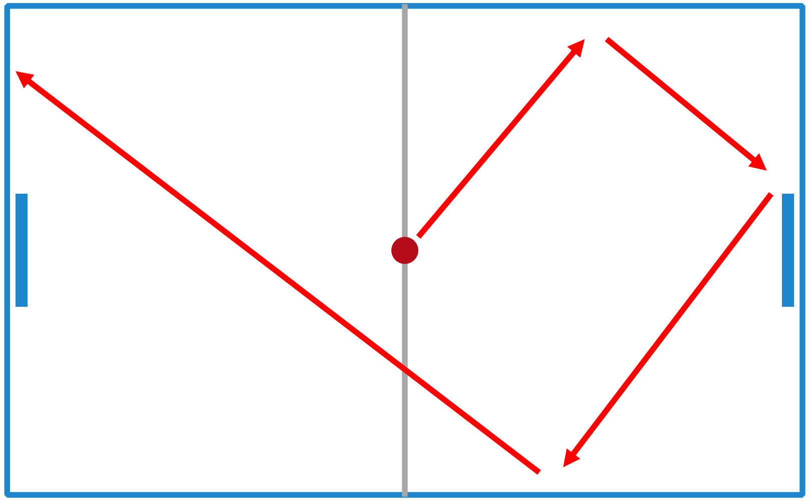
- ❑ `println( (a > b) ? “smaller” : “bigger” );`
- ❑ `println( (a==b) ? “bingo” : “try again” );`

Exercise: Mouse Drawing

```
void setup() {
  size(300, 300);
  background(255);
}

void draw() {
  if (mousePressed){ //draw a line while mouse pressed
    stroke(255,0,0);
    line(pmouseX,pmouseY, mouseX,mouseY);
  }
}

// TODO: use mouseButton to change line color
// LEFT button: red , other buttons: black
```

Exercise1: Pong game

- ❑ Drawing a Bouncing Ball
- ❑ Detect the boundaries of the canvas and updating the position of the ball as it bounces off the walls
- ❑ Hints:
  using logical operators



Interrupted by Event(偵測輸入事件)

```
void setup() {
  size(300, 300);
  background(255);
}


void draw() {
}
```

triggered by mouse event

```
void mousePressed() {
  ellipse(mouseX,mouseY,10,10);
}
```

triggered by keyboard event

```
void keyPressed() {
  background(255);
}
```

Exercise2

- ❑ Press a key to toggle the animation
- ❑ Use mouse to control vertical position of the paddle
- ❑ Hit detection: detecting collisions between the paddles and the ball
- ❑ Print scores and remaining lives
  - ❑ +10/hit, lost a life for not hitting the ball

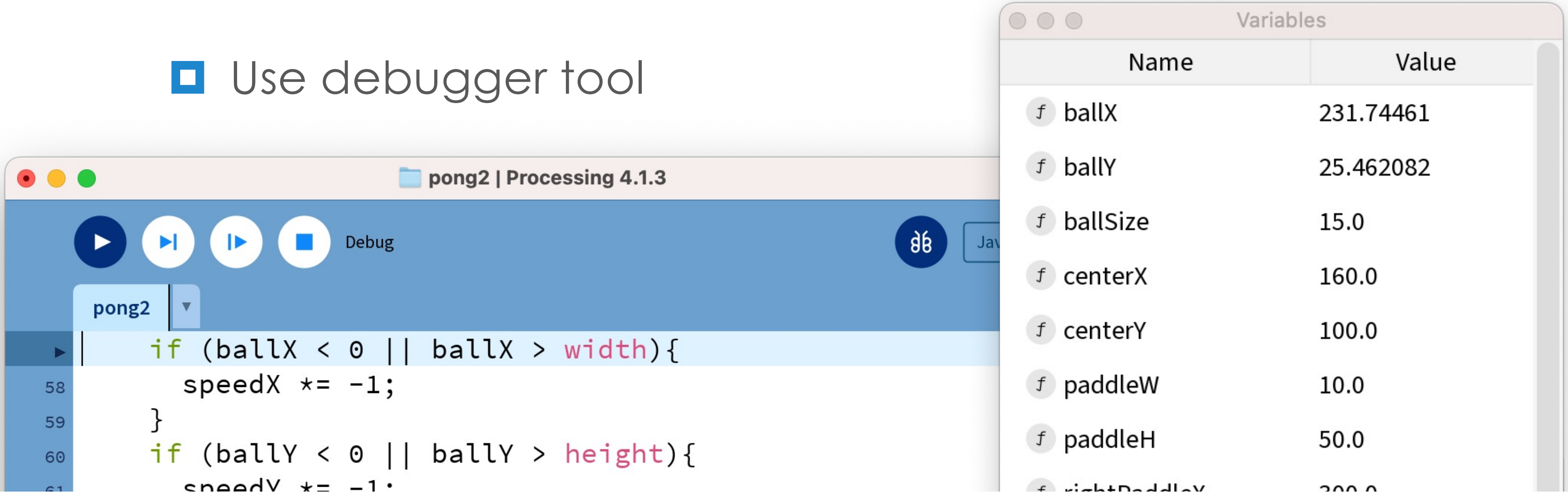
Types of bugs

- Compile-time errors
  - syntax error, type error…
- Runtime exceptions
  - dividing by zero, null pointer…
- Logic flaws
  - the program does not behave as intended

http://stackoverflow.com

Debugging process

- break down your problem into smaller pieces
  - Use // or /* */ to temporarily disable parts of your code
- use println() to check variable values in the Console window
- Use debugger tool



```
if (ballX < 0 || ballX > width){
  speedX *= -1;
}
if (ballY < 0 || ballY > height){
  speedY *= -1;
```

Name	Value
ballX	231.74461
ballY	25.462082
ballSize	15.0
centerX	160.0
centerY	100.0
paddleW	10.0
paddleH	50.0
rightPaddleX	300.0

Recap

- Data types
- Type conversions
- Constants
- Conditionals and boolean expressions
- Boundary detection, Hit detection, Key and mouse events
- Debugging process