



THE UNIVERSITY OF
MELBOURNE

COMP90041

Programming and Software Development

Tutorial 3 Flow of Control

Chuang(Frank) Wang

Slides were developed by Chuang Wang
Copyright @ The University of Melbourne





Overview

1. Execution Flow
2. Branching
3. Loops
4. Exercises & Lab



THE UNIVERSITY OF
MELBOURNE

1. Execution Flow



Example

```
1 printThreeThings(2, 3, 4);
```

```
1 public void printThreeThings(int x, int y, int z){  
2     System.out.println(x);  
3     System.out.println(y);  
4     System.out.println(z);  
5 }
```



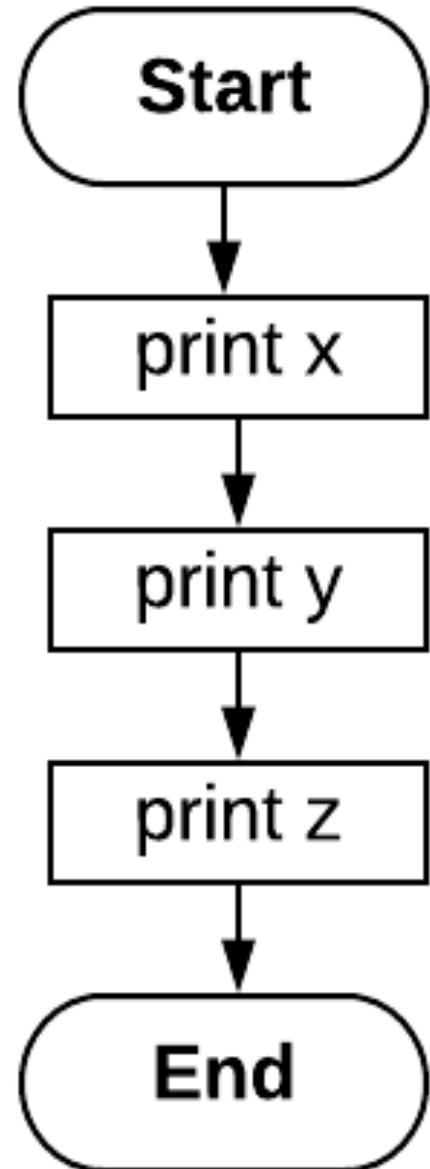
Flowchart Symbols

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

Execution Flow

```
1 printThreeThings(2, 3, 4);
```

```
1 public void printThreeThings(int x, int y, int z){  
2     System.out.println(x);  
3     System.out.println(y);  
4     System.out.println(z);  
5 }
```





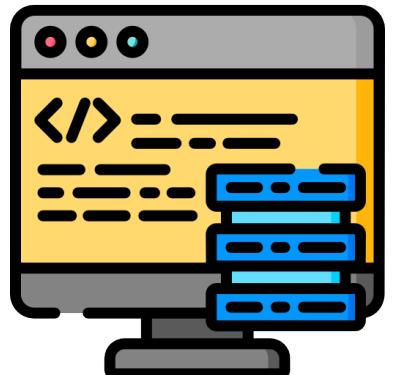
THE UNIVERSITY OF
MELBOURNE

2. Branching



Branching

- **if-else Statement**
- **switch Statement**





if-else Statement

```
if (Boolean_Expression)
```

```
    Yes_Statement
```

```
else
```

```
    No-Statement
```



switch Statement

WHY?

```
switch(Controlling_Expression){  
    case Case_Label_1:  
        Statement_1;  
        break;  
    case Case_Label_2:  
        Statement_2;  
        break;  
    default:  
        default_Statement;  
        break;  
}
```



THE UNIVERSITY OF
MELBOURNE

3. Loops



Difference?

While loop

```
while (Boolean_Expression) {  
    Statement 1;  
    Statement 2;  
    :  
    Statement last;  
}
```

do-while loop

```
do {  
    Statement 1;  
    Statement 2;  
    :  
    Statement last;  
} while (Boolean_Expression);
```



for loop

```
for (initialization; Boolean; Update) {  
    Body  
}
```



THE UNIVERSITY OF
MELBOURNE

4. Exercise

if-else Statement

```
public void printIfThree(int i) {  
    if (i == 3) {  
        System.out.println("yay, it's three");  
    } else {  
        System.out.println("not 3!");  
        System.out.println(i);  
    }  
    System.out.println("done");  
}
```

for loop

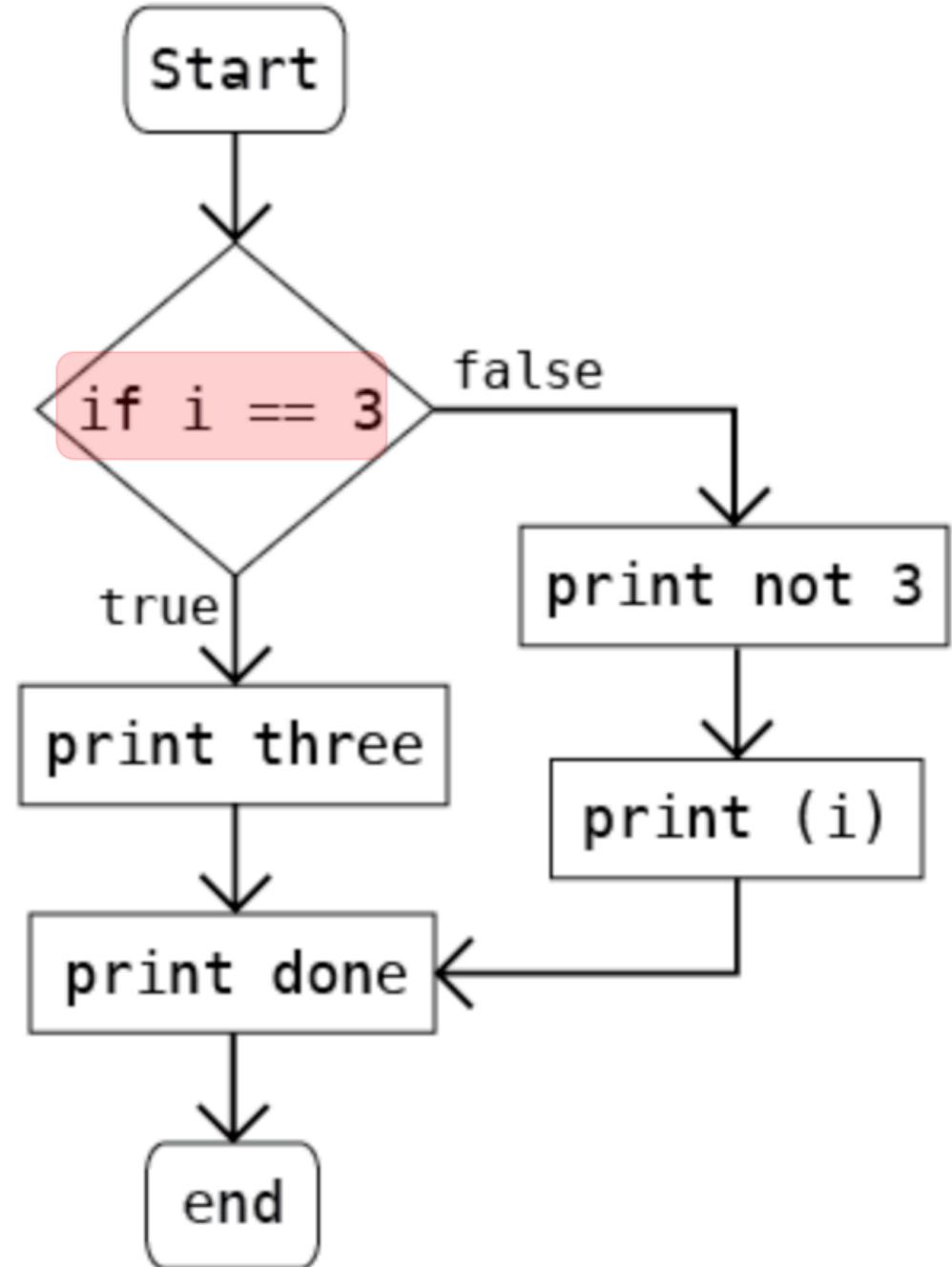
```
public void printNumbers(int x){  
    for (int i = 0; i < x; i++){  
        System.out.println(i);  
    }  
}
```



if-else Statement

Boolean

```
public void printIfThree(int i) {  
    if (i == 3) {  
        System.out.println("yay, it's three");  
    } else {  
        System.out.println("not 3!");  
        System.out.println(i);  
    }  
    System.out.println("done");  
}
```



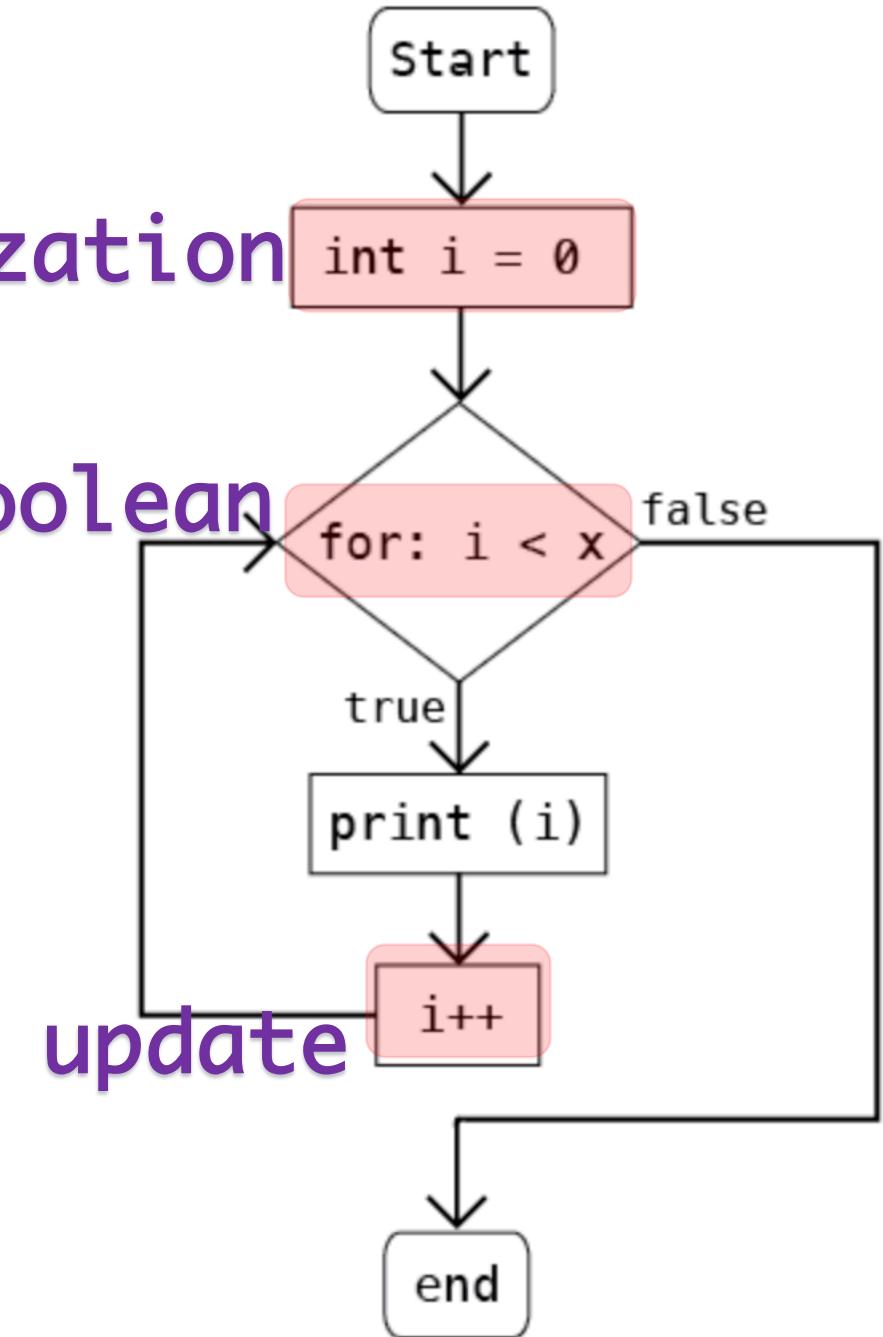


for Loop

```
public void printNumbers(int x){  
    for (int i = 0; i < x; i++){  
        System.out.println(i);  
    }  
}
```

initialization

Boolean



update



Tutorial Exercise

1. Write a program that takes one command line argument, which should be the abbreviation of a compass direction, that is, one of N, S, E, or W. The program should print out number of degrees east of north that are specified by that heading. So for example, if the input is N, it should print out 0, and if it is W, it should print out 270. If the input is not a valid compass direction abbreviation, print out a suitable error message and exit the program.

Hint: If you are using Java 7 or later, this is easier, as Java 7 introduced switching on strings. Your IDE may allow you to specify which Java version it supports, or you can execute `java -version` on the command line to check the version. If it says version 1.7 or more, you can switch on strings.

Hint: If `s` is a string, `s.charAt(0)` will return the first character of the string. Every Java version allows you to switch on characters.

Hint: You can exit any program immediately in an error state with the statement

```
System.exit(1);
```

If you replace the 1 with a 0, the statement is appropriate for exiting immediately in a success state, as would happen if you reached the end of the `main` method normally.



Homework

*	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

2. Write a program to print a 10×10 multiplication table. Print the numbers from 1 to 10 across the top and down the left side, and show the products in the middle. Make sure your columns line up neatly. Print a heading row at the top showing the numbers from 1 to 10, and a leftmost column also showing the numbers from 1 to 10. To make it look nicer, print a | character between the columns of each row, and a solid row of - character between the rows. To make it look even neater, print a + character wherever the vertical and horizontal lines meet.



Thank you





WARNING

This material has been reproduced and communicated to you by or on behalf of the University of Melbourne in accordance with section 113P of the *Copyright Act 1968 (Act)*.

The material in this communication may be subject to copyright under the Act.

Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice