COMP 110-001
Flow of Control: Branching 1

Yi Hong

May 19, 2015

Today

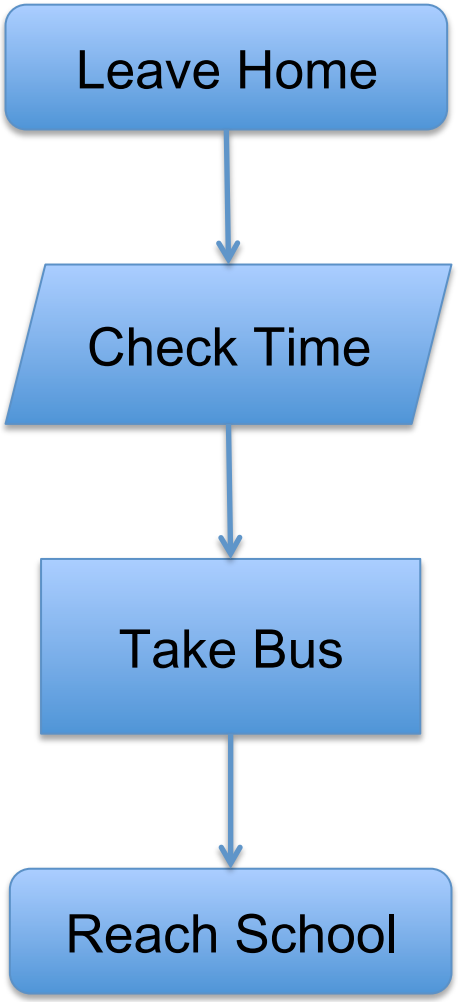
- The if-else statement
- Boolean expressions

Flow of Control

- The order in which a program performs actions
  - Continuation (unconditional)
    - Until now, actions are taken sequentially
  - More complicated flow of control:
    - A **branching statement** chooses an action from a list of two or more possible actions
    - A **loop statement** repeats an action again and again until some stopping condition is met

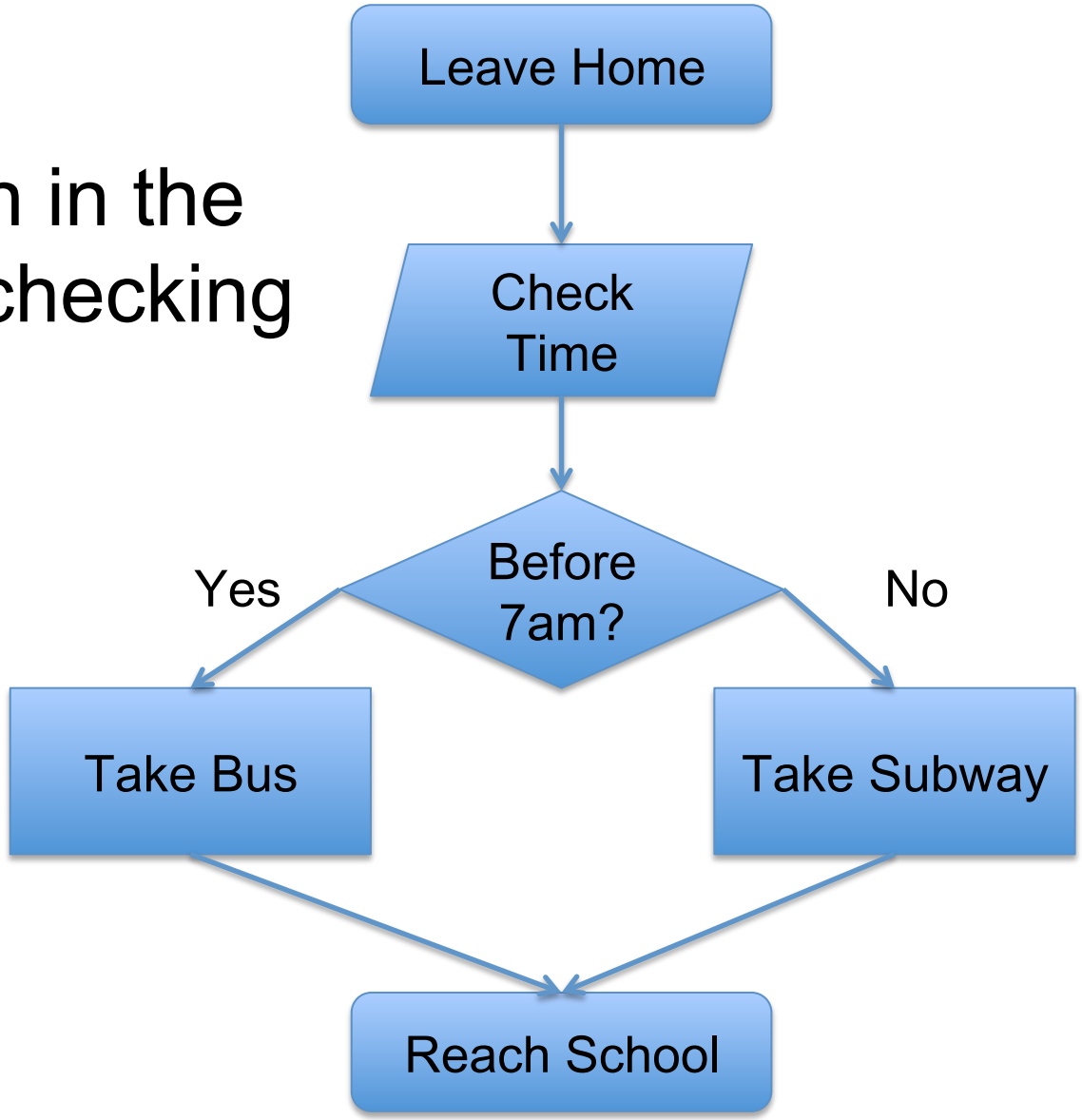
Simple order

- Continuation (unconditional)
- Perform actions sequentially
- A single path in the flow chart



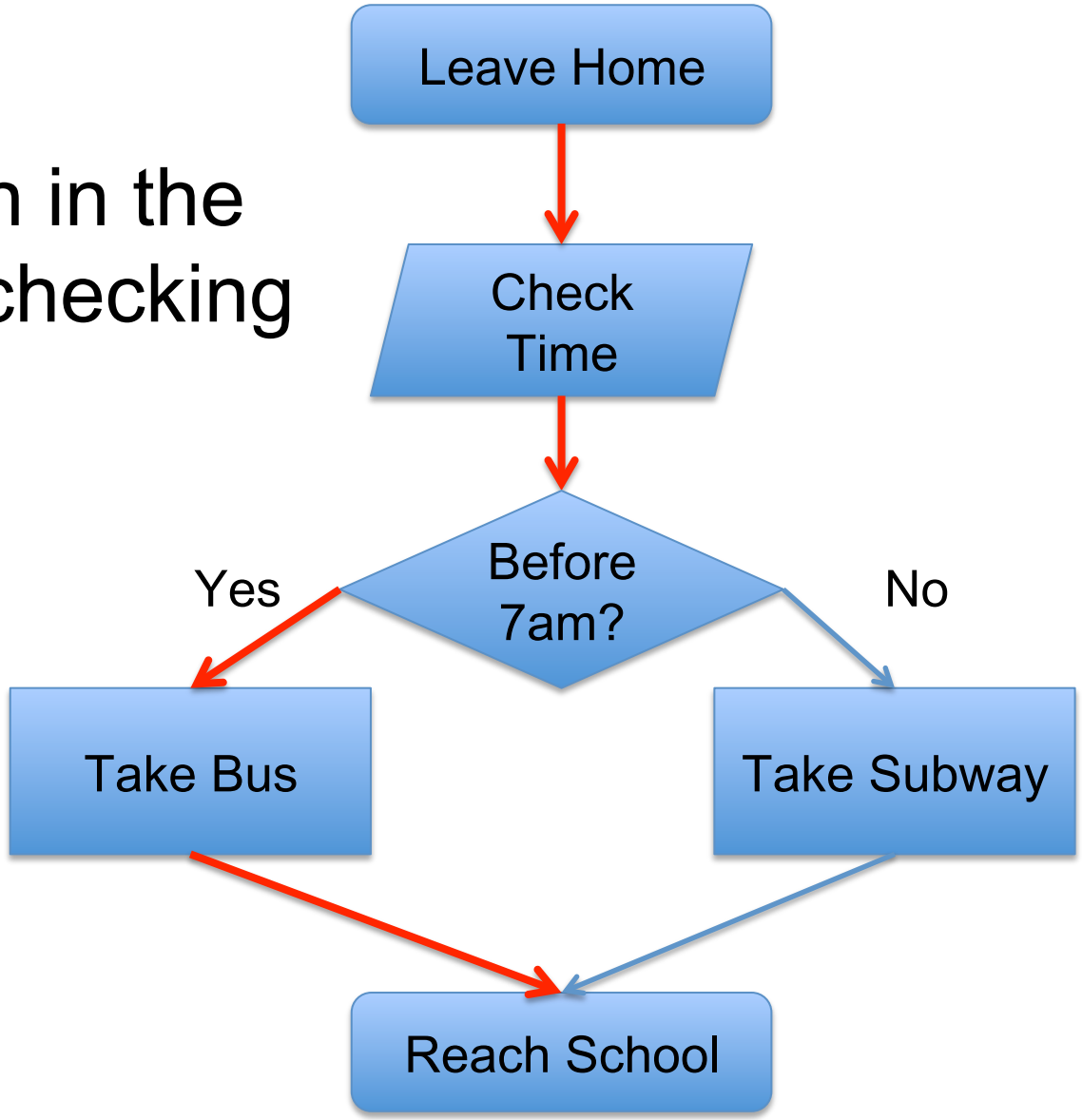
Branching

- Choose a path in the flow chart by checking conditions



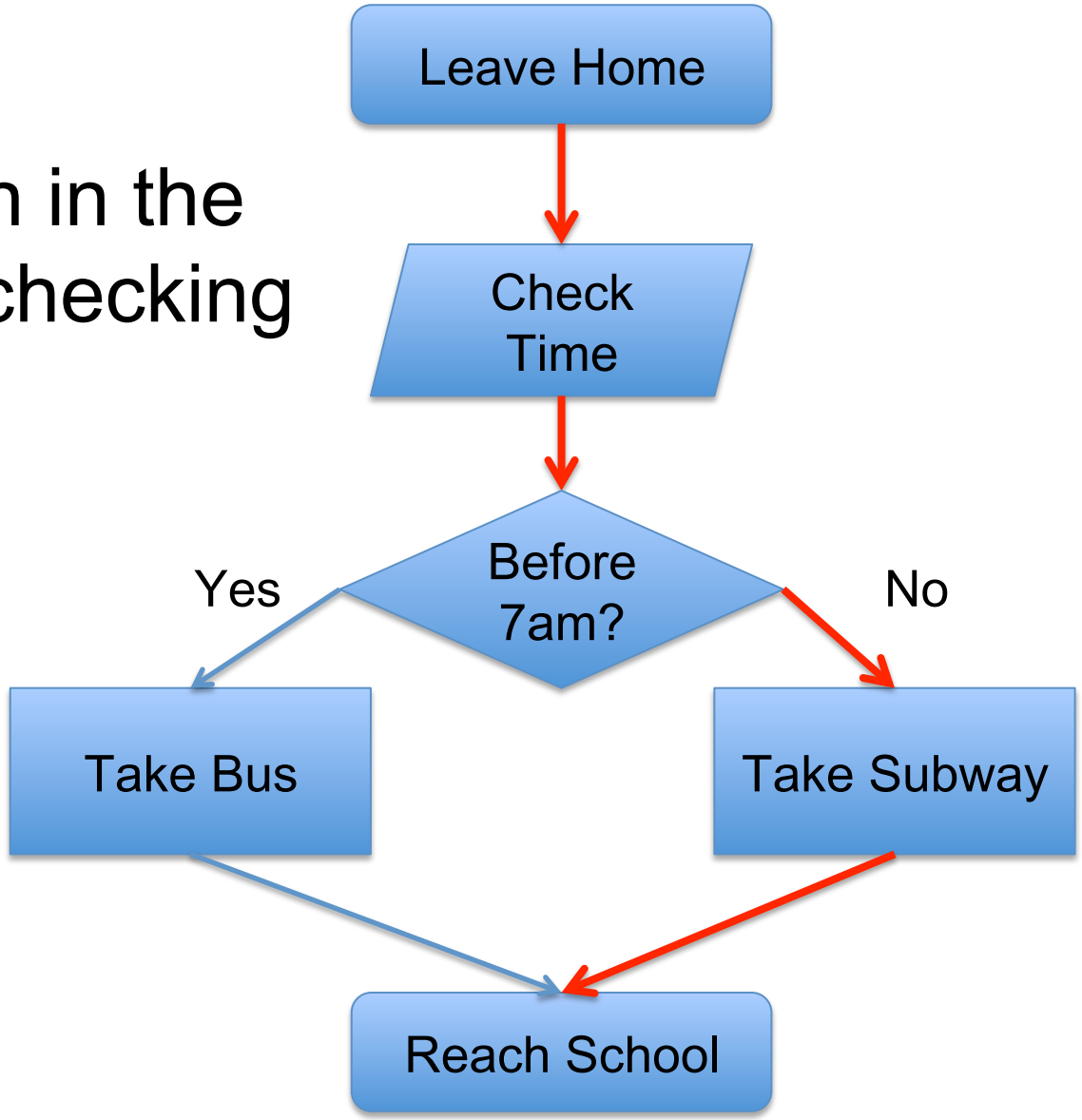
Branching

- Choose a path in the flow chart by checking conditions



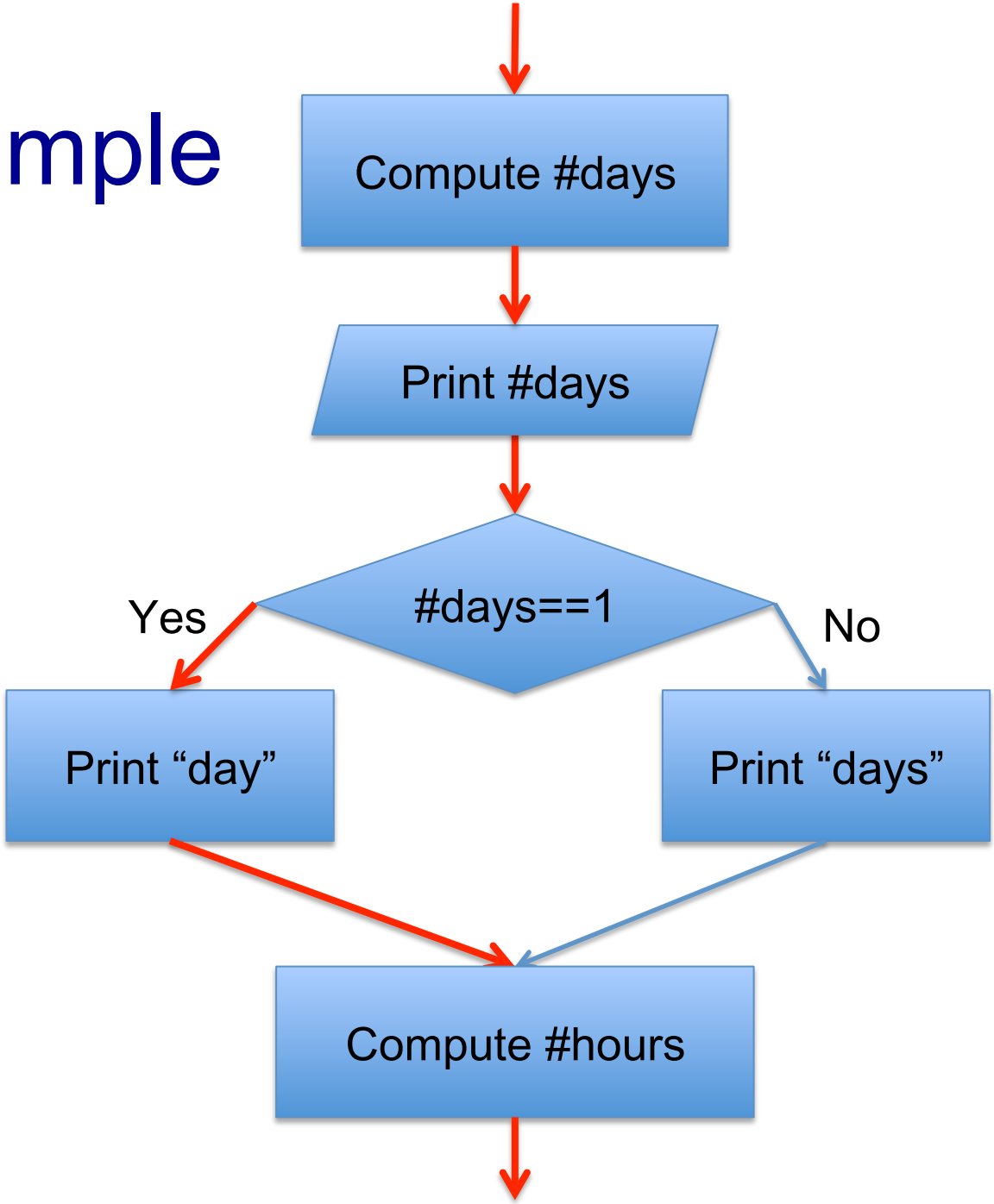
Branching

- Choose a path in the flow chart by checking conditions



Another Example

- In homework 1



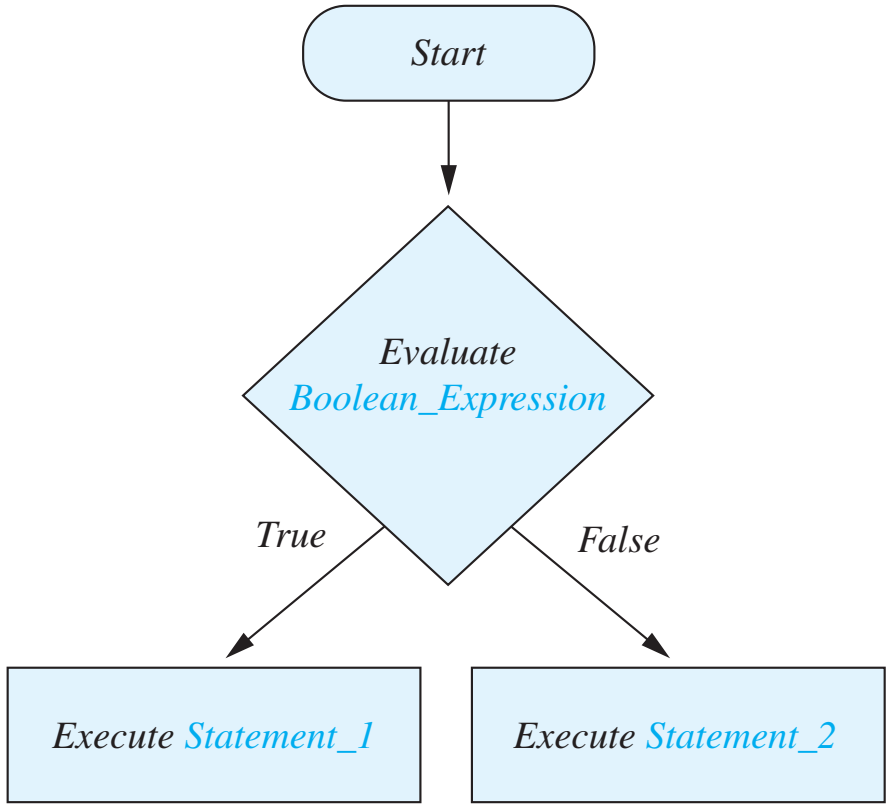
The if-else Statement

- A branching statement that chooses between two possible actions
- Syntax

```
if (Boolean_Expression)
   Statement_1
else
   statement_2
```

Example:

```
if (days == 1)
   system.out.print(“ day”);
else
   System.out.print(“ days”);
```

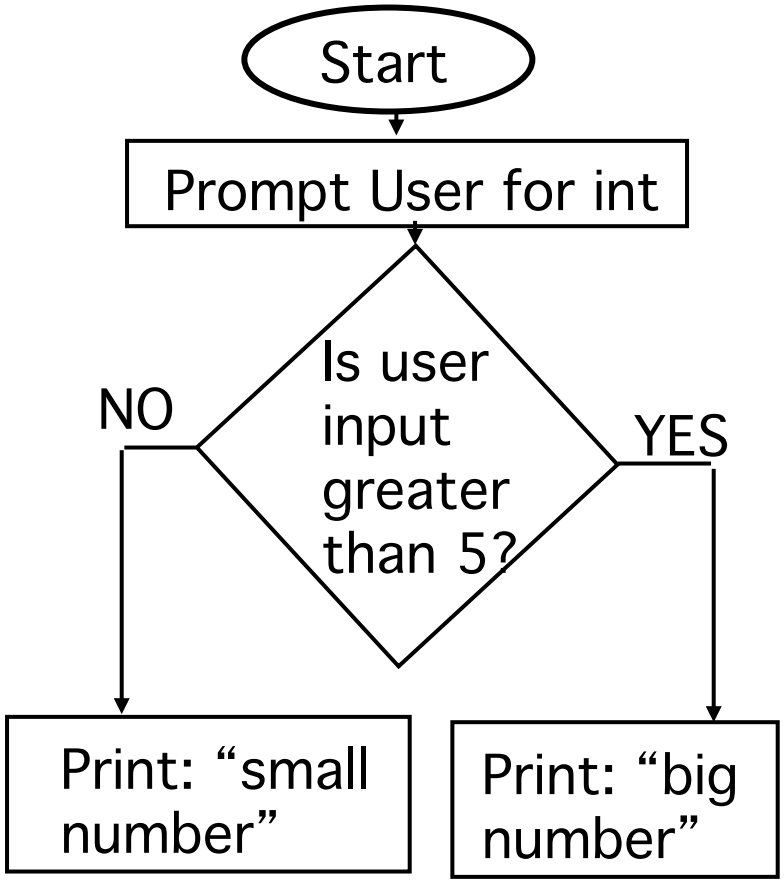


The if-else Statement

- What if you have multiple statements in each branch?

```
if (Boolean_Expression){
  Statement_1.1
  Statement_1.2
  ...
}
else {
  Statement_2.1
  Statement_2.2
  ...
}
```

Java Example




```
import java.util.*;

public class FlowChart
{
   public static void main(String[] args)
   {
      System.out.println("Give me an integer:");
      Scanner keyboard = new Scanner(System.in);
      int inputInt = keyboard.nextInt();

      if( inputInt  >  5)
      {
         System.out.println("Big number");
      }
      else
      {
         System.out.println("Small number");
      }
   }
}
```

Java Comparison Operators

==	Equal to
!=	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

Boolean Expressions

- if (boolean_expression) { statements }
- True or false
- “atomic” expressions

```
5 == 3              // always false
myInt <= 6          // depends on the value of myInt

// depends on the value of myInt, anotherInt
myInt != anotherInt
```

Review of Boolean Operators

- The effect of the boolean operators && (and), || (or), and ! (not) on boolean values

Value of *A*	Value of *B*	Value of *A* && *B*	Value of *A* || *B*	Value of !(*A*)
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

Use && for and

- Form a larger boolean expression out of many smaller boolean expressions
- Syntax
  - (Sub_Expression_1) && (Sub_Expression_2) && …
- Will be true if and only if ALL statements are true

Use || for or

- Also form a larger boolean expression out of many boolean expressions
- Syntax
  - (Sub_Expression_1) || (Sub_Expression_2) || …
- Will be true if **ONE** expression is true

Example in Homework 1

- Check if #days is not 1
  - days ! = 1
  - !(days == 1)
  - days < 1 || days > 1
  - !(days >= 1 && days <= 1)

Comparison for Objects

- Call object's method equals() method
- E.g.: Compare two strings
  - str1 = str2; (✗ assignment statement)
  - str1 == str2; (Do NOT, == tests whether they are stores in the same memory location)
  - str1.equals(str2); (GOOD)

Compare Strings

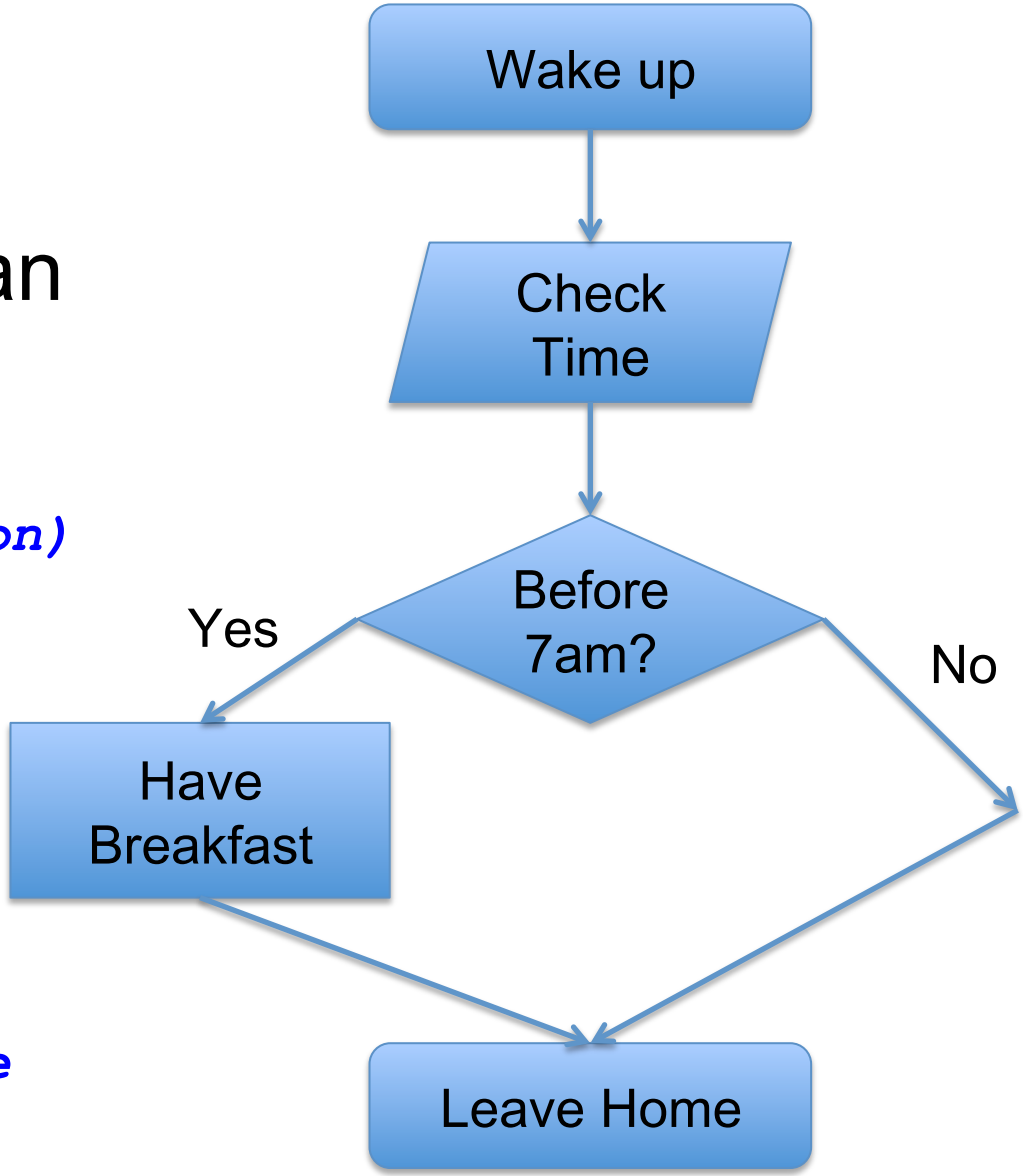
- Syntax
  - String.equals(Other_String)
  - String.equalsIgnoreCase(Other_String)
- Example

```
String name1 = “COMP110”;
String name 2 = new String(“COMP110”);
if (name1.equals(name2)){
      System.out.println(“The same”);
} else {
      System.out.println(“Different”);
}
```

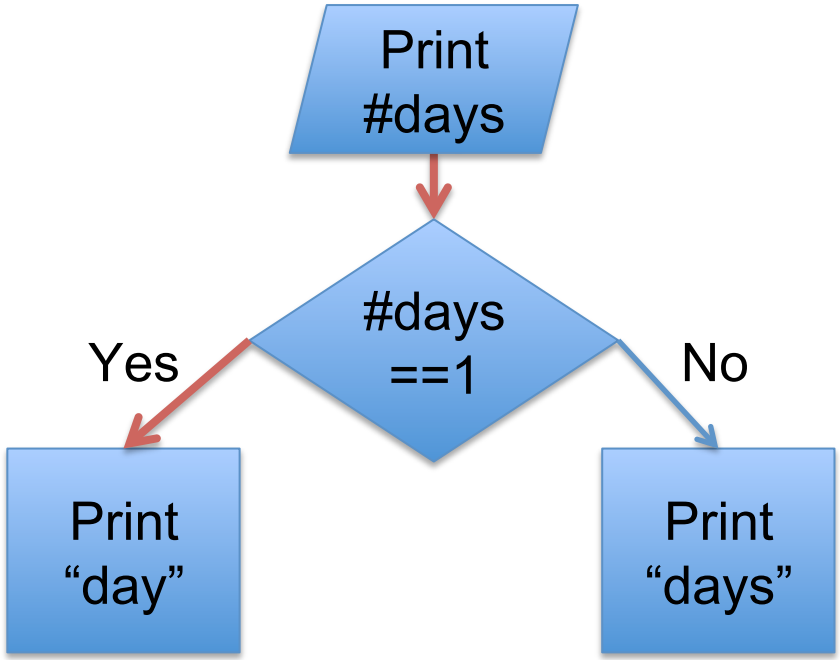
If Without Else

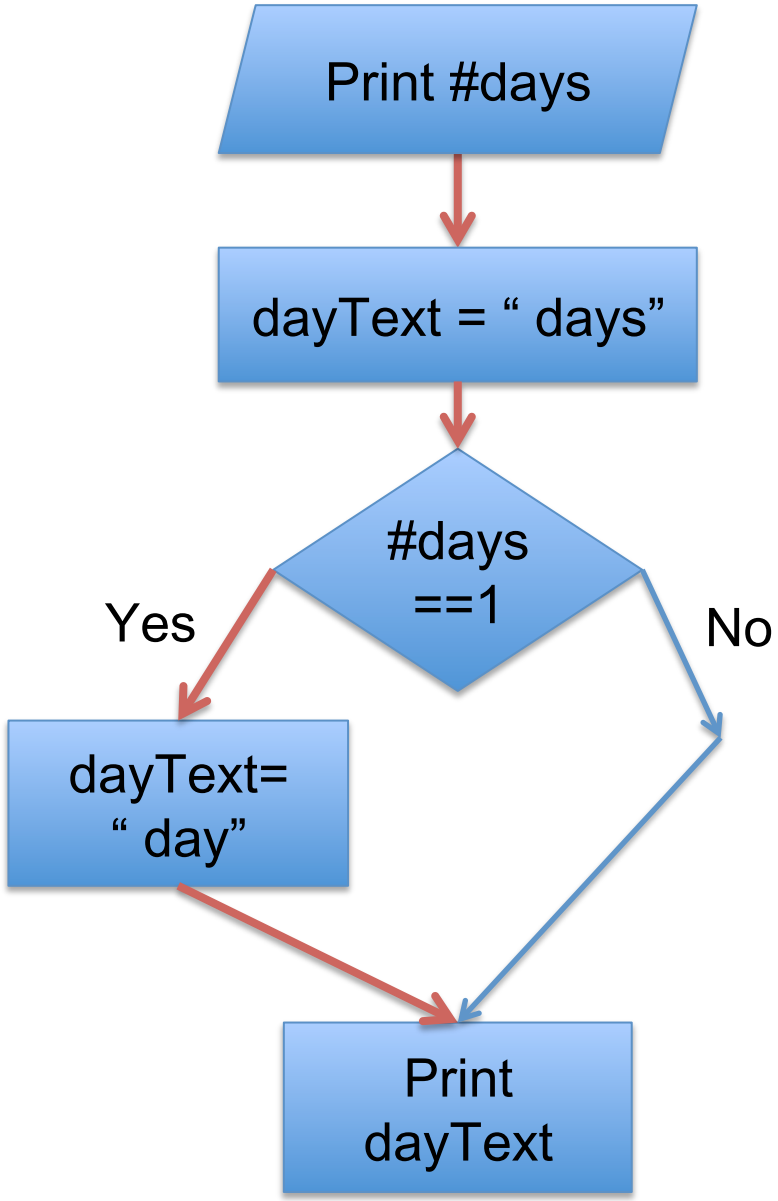
- You can use just an if statement

```
if (Boolean_Expression)
{
  Statement_1.1
  Statement_1.2
  ...
}
the rest of your code
```



A Different Implement for Homework 1

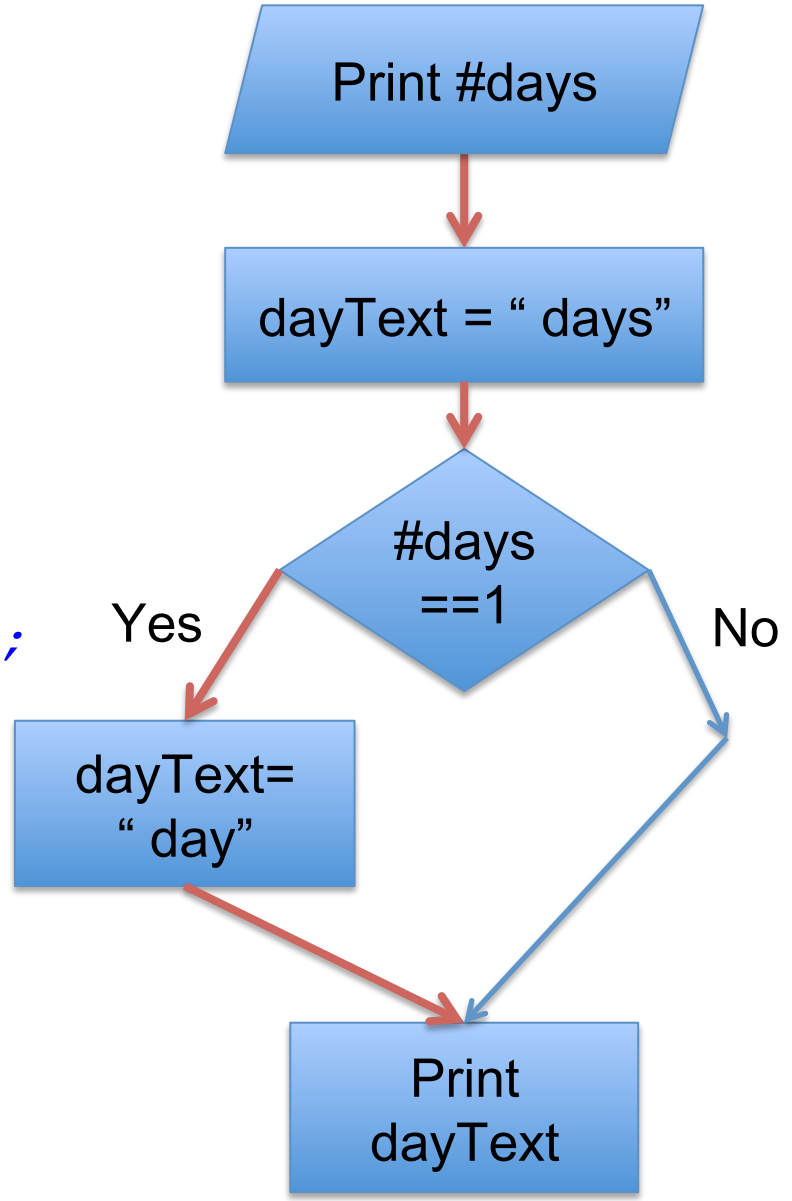




A Different Implement for Homework 1

```
System.out.print( days );
String dayText = " days";

if (days==1){
    dayText = " day";
}
System.out.print( dayText );
```



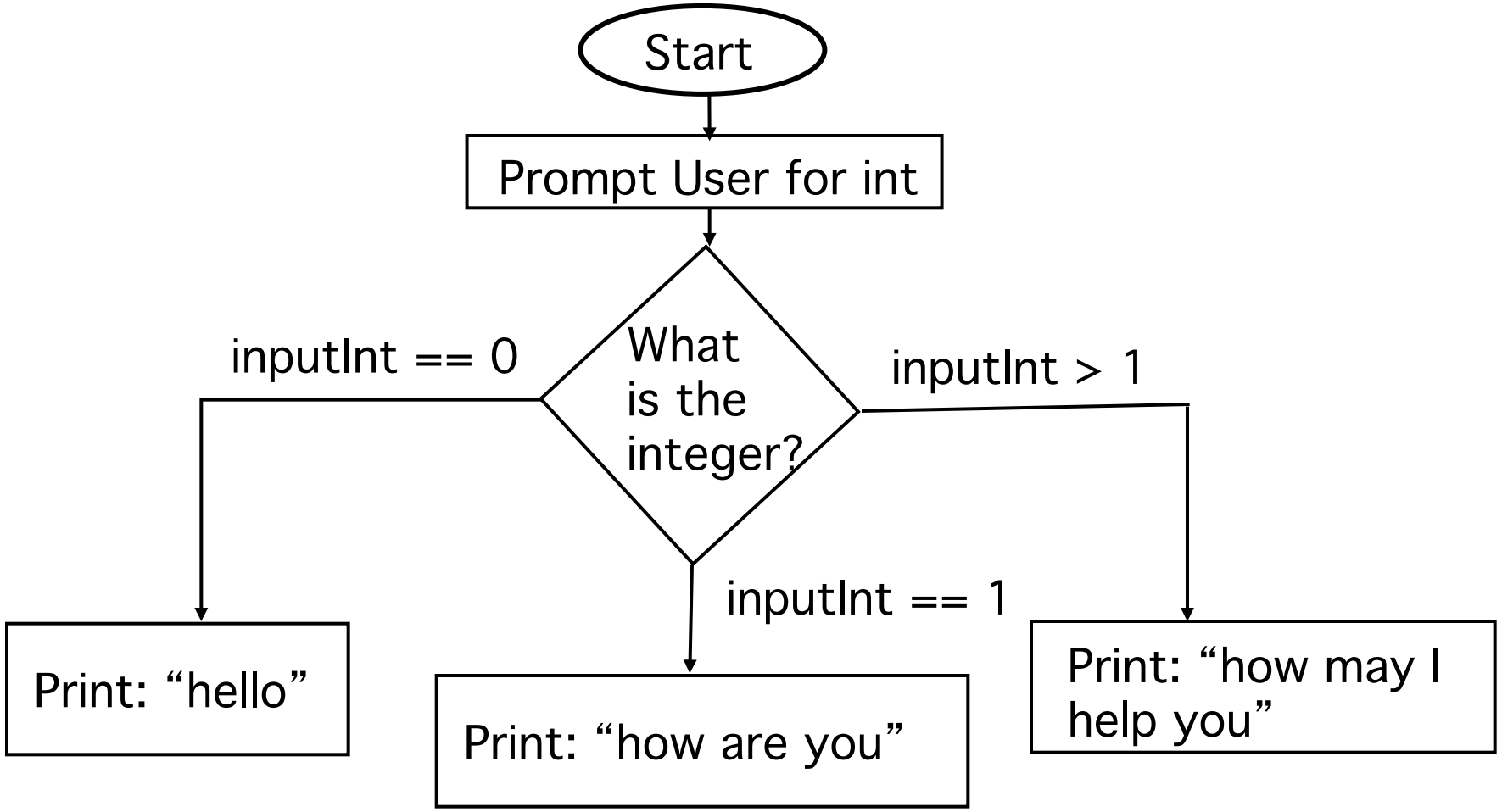
Nested if-else Statements

- An if-else statement can contain any sort of statements within it
- You can nest an if-else statement within another if-else statement

```
if (boolean expression)
{
      if (boolean expression){
            stuff goes here
      } else {
      more stuff }
} else {
…
}
```

```
if (boolean expression)
{
      …
} else if (boolean expression){
      stuff goes here
}
else {
      more stuff
}
```

Pseudocode in Flowchart Format



```
import java.util.*;

  public class FlowChart
  {
    public static void main(String[] args)
    {
      System.out.println("Give me an integer:");
      Scanner keyboard = new Scanner(System.in);
      int inputInt = keyboard.nextInt();

      if ( inputInt  ==  0)
        System.out.println(”hello");
      else if ( inputInt == 1)
        System.out.println(”how are you");
      else if (inputInt > 1)
        System.out.println(”how may I help you");
      else
        System.out.println(“Negative”);
  }
```

Next Class

- More if / else statements
- The switch statements