

This lesson is on control structure branching

OVERVIEW

The following are the coverage for Control structures -

Branching

- Relational and Logical Operators
- if, if-else, if-else if-else Statement
- Nested if Statement
- The switch Statement
- Conditional Operator

Content Copyright Nanyang Technological University

2

Basic C Programming

There are 5 main sections to cover for Control structures (branching).

This video lesson focuses on the second part: if, if...else, if...else if...else....else statements

LEARNING OBJECTIVES

Content Copyright Nanyang Technological University 3

LEARNING OBJECTIVES

After this lesson, you should be able to:

Content Copyright Nanyang Technological University 4

After this lesson, you should be able to:

LEARNING OBJECTIVES

After this lesson, you should be able to:

- Explain how if statement works with example

Content Copyright Nanyang Technological University

5

- Explain how if statement works with example

LEARNING OBJECTIVES

After this lesson, you should be able to:

- Explain how if statement works with example
- Explain how if–else statement works with example

Content Copyright Nanyang Technological University 6

- Explain how if –else statement works with example

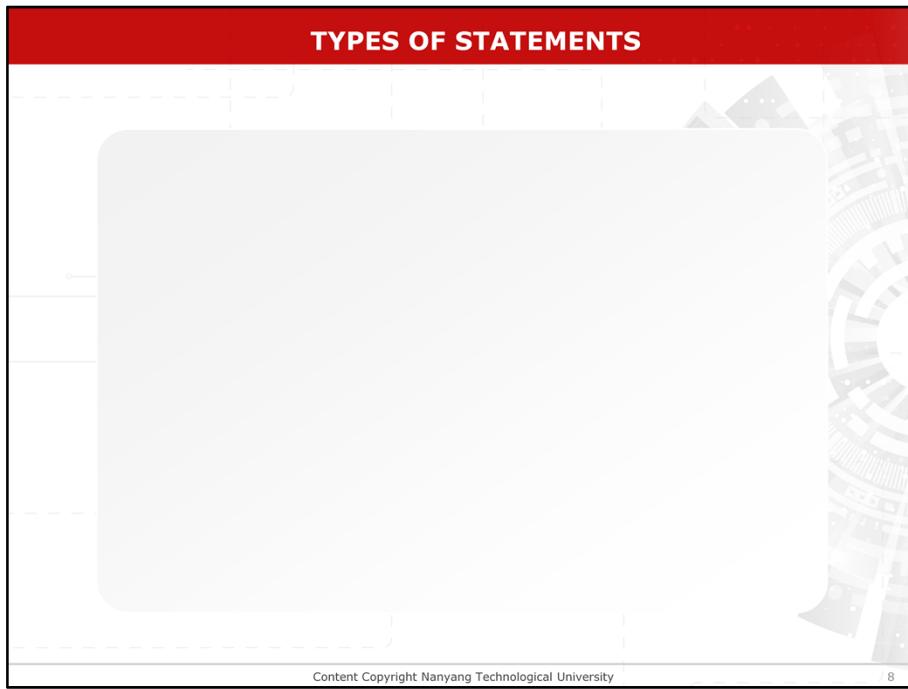
LEARNING OBJECTIVES

After this lesson, you should be able to:

- Explain how if statement works with example
- Explain how if-else statement works with example
- Explain how if-else if-else statement works with example

Content Copyright Nanyang Technological University 7

- Explain how if-else if-else statement works with example



The if Statement

The C language has two types of statements for implementing the branching control structure:

TYPES OF STATEMENTS

- The C language has two types of statements for implementing the branching control structure:

Content Copyright Nanyang Technological University

9

The if Statement

The C language has two types of statements for implementing the branching control structure:

TYPES OF STATEMENTS

- The C language has two types of statements for implementing the branching control structure:
 - if statement

Content Copyright Nanyang Technological University 10

The if Statement

The C language has two types of statements for implementing the branching control structure: **if** statement

TYPES OF STATEMENTS

- The C language has two types of statements for implementing the branching control structure:
 - if statement
 - switch statement

Content Copyright Nanyang Technological University 11

The if Statement

The C language has two types of statements for implementing the branching control structure: **if** statement and **switch** statement.

TYPES OF STATEMENTS

- The C language has two types of statements for implementing the branching control structure:
 - if statement
 - switch statement
- The if statement can be used for two-way selection, and the nested-if statement can be used for multi-way selection.

Content Copyright Nanyang Technological University 12

The **if** statement can be used for two-way selection, and the nested-**if** statement can be used for multi-way selection.

TYPES OF STATEMENTS

- The C language has two types of statements for implementing the branching control structure:
 - if statement
 - switch statement
- The if statement can be used for two-way selection, and the nested-if statement can be used for multi-way selection.
- The switch statement is used for multi-way selection.

Content Copyright Nanyang Technological University 13

The **switch** statement is used for multi-way selection.

The slide has a red header bar with the title 'THE IF STATEMENT'. The main content area contains the following text in a white box:

```
if (expression)
    statement;
/* simple or compound statement
   enclosed with brackets { } */
```

At the bottom left of the slide, there is a small text: 'Content Copyright Nanyang Technological University'. At the bottom right, there is a small number '14'.

The simplest form of the **if** statement is shown here.

The slide has a red header bar with the title 'THE IF STATEMENT'. Below the title is a code block containing the syntax for an if statement:

```
if (expression)
    statement;
/* simple or compound statement
   enclosed with brackets { } */
```

At the bottom left of the slide, there is a small text: 'Content Copyright Nanyang Technological University'. At the bottom right, there is a small number '15'.

where **if** is a reserved keyword.

THE IF STATEMENT

```
if (expression)
    statement;
/* simple or compound statement
   enclosed with brackets { } */
```

The flowchart illustrates the execution of an if statement. It starts with an input box labeled "expression". An arrow points from "expression" to a decision diamond labeled "true" and "false". If the expression is true, an arrow points down to a box labeled "statement", which then leads to an output circle. If the expression is false, the flow bypasses the "statement" box and continues directly to the output circle.

If the **expression** is evaluated to be true (i.e. non-zero), then the **statement** is executed. If the **expression** is evaluated to be false (i.e. zero), then the **statement** is ignored, and the control is passed to the next program statement following the **if** statement.

The slide has a red header bar with the title "THE IF STATEMENT". The main content area contains the following text:

```
if (expression)
    statement;
/* simple or compound statement
   enclosed with brackets { } */
```

At the bottom left of the slide, there is a small text: "Content Copyright Nanyang Technological University". At the bottom right, there is a small number: 17.

The **statement** may be a single statement terminated by a semicolon

THE IF STATEMENT

```
if (expression)
    statement;
/* simple or compound statement
   enclosed with brackets { } */

if (expression) {
    statement1;
    statement2;
    ...
    statmentn;
}
```

Content Copyright Nanyang Technological University 18

or it may be a compound statement enclosed by braces.

THE IF STATEMENT

```
/* Program: check user number greater than 5 */
#include <stdio.h>
int main()
{
    int num;
    printf("Give me a number from 1 to 10:");
    scanf("%d", &num);
    if (num > 5)
        printf("Your number is larger than 5.\n");
    printf("%d was the number you entered.\n", num);
    return 0;
}
```

Content Copyright Nanyang Technological University

19

In the example program, the purpose is to read in an user number and check whether the number is greater than 5.

THE IF STATEMENT

```
/* Program: check user number greater than 5 */
#include <stdio.h>
int main()
{
    int num;
    printf("Give me a number from 1 to 10:");
    scanf("%d", &num);
    if (num > 5)
        printf("Your number is larger than 5.\n");
    printf("%d was the number you entered.\n", num);
    return 0;
}
```

Content Copyright Nanyang Technological University 20

The program asks the user to enter a number from 1 to 10

THE IF STATEMENT

```
/* Program: check user number greater than 5 */
#include <stdio.h>
int main()
{
    int num;
    printf("Give me a number from 1 to 10: ");
    scanf("%d", &num);
    if (num > 5)
        printf("Your number is larger than 5.\n");
    printf("%d was the number you entered.\n", num);
    return 0;
}
```

Output

Give me a number
from 1 to 10: 3

Content Copyright Nanyang Technological University 21

The program asks the user to enter a number from 1 to 10.

THE IF STATEMENT

```
/* Program: check user number greater than 5 */
#include <stdio.h>
int main()
{
    int num;
    printf("Give me a number from 1 to 10: ");
    scanf("%d", &num);
    if (num > 5)
        printf("Your number is larger than 5.\n");
    printf("%d was the number you entered.\n", num);
    return 0;
}
```

Output

Give me a number
from 1 to 10: 3

3 was the number
you entered.

Content Copyright Nanyang Technological University 22

and then reads the user input.

THE IF STATEMENT

```
/* Program: check user number greater than 5 */
#include <stdio.h>
int main()
{
    int num;
    printf("Give me a number from 1 to 10: ");
    scanf("%d", &num);
    if (num > 5)
        printf("Your number is larger than 5.\n");
    printf("%d was the number you entered.\n", num);
    return 0;
}
```

Output

Give me a number
from 1 to 10: 3

3 was the number
you entered.

Content Copyright Nanyang Technological University 23

The expression in the **if** statement contains a relational operator. It checks to see whether the user input is greater than 5.

THE IF STATEMENT

```
/* Program: check user number greater than 5 */  
#include <stdio.h>  
int main()  
{  
    int num;  
    printf("Give me a number from 1 to 10: ");  
    scanf("%d", &num);  
    if (num > 5)  
        printf("Your number is larger than 5.\n");  
    printf("%d was the number you entered.\n", num);  
    return 0;  
}
```

Output

Give me a number
from 1 to 10: 7

Content Copyright Nanyang Technological University 24

Otherwise, the **printf()** statement will print the string

THE IF STATEMENT

```
/* Program: check user number greater than 5 */
#include <stdio.h>
int main()
{
    int num;
    printf("Give me a number from 1 to 10: ");
    scanf("%d", &num);
    if (num > 5)
        printf("Your number is larger than 5.\n");
    printf("%d was the number you entered.\n", num);
    return 0;
}
```

Output

Give me a number from
1 to 10: 7

Your number is larger
than 5.

Content Copyright Nanyang Technological University 25

"Your number is larger than 5." on the screen.

THE IF STATEMENT

```
/* Program: check user number greater than 5 */
#include <stdio.h>
int main()
{
    int num;
    printf("Give me a number from 1 to 10: ");
    scanf("%d", &num);
    if (num > 5)
        printf("Your number is larger than 5.\n");
    printf("%d was the number you entered.\n", num);
    return 0;
}
```

Output

Give me a number
from 1 to 10: 7

Your number is
larger than 5.

7 was the number
you entered.

Content Copyright Nanyang Technological University 26

Finally, the last **printf()** statement will be executed to print the number entered by the user on the screen.

THE IF-ELSE STATEMENT

Content Copyright Nanyang Technological University 27

The if-else Statement

The **if-else** statement implements a two-way selection. The format of the **if-else** statement is

```
if (expression)
    statement1;
else
    statement2;
```

where **if** and **else** are reserved keywords. When the **if-else** statement is executed, the **expression** is evaluated. If **expression** is true, then **statement1** is executed and the control is passed to the program statement following the **if** statement. If **expression** is false, then **statement2** is executed. Both **statement1** and **statement2** may be a single statement terminated by a semicolon or a compound statement enclosed by {}.

In the example program, the purpose is to read in two integer numbers and determines the maximum number between them.

The slide has a red header bar with the title "THE IF-ELSE STATEMENT". Below the title is a code block containing the pseudocode for an if-else statement. The code is as follows:

```
if (expression)
    statement1;
else
    statement2;
```

At the bottom left of the slide, there is a small text "Content Copyright Nanyang Technological University" and at the bottom right, the number "28".

The if-else Statement

The **if-else** statement implements a two-way selection. The format of the **if-else** statement is as shown.

The slide has a red header bar with the title "THE IF-ELSE STATEMENT". Below the title is a code block containing the following pseudocode:

```
if (expression)
    statement1;
else
    statement2;
```

At the bottom left of the slide, there is a small text: "Content Copyright Nanyang Technological University". At the bottom right, there is a small number: "29".

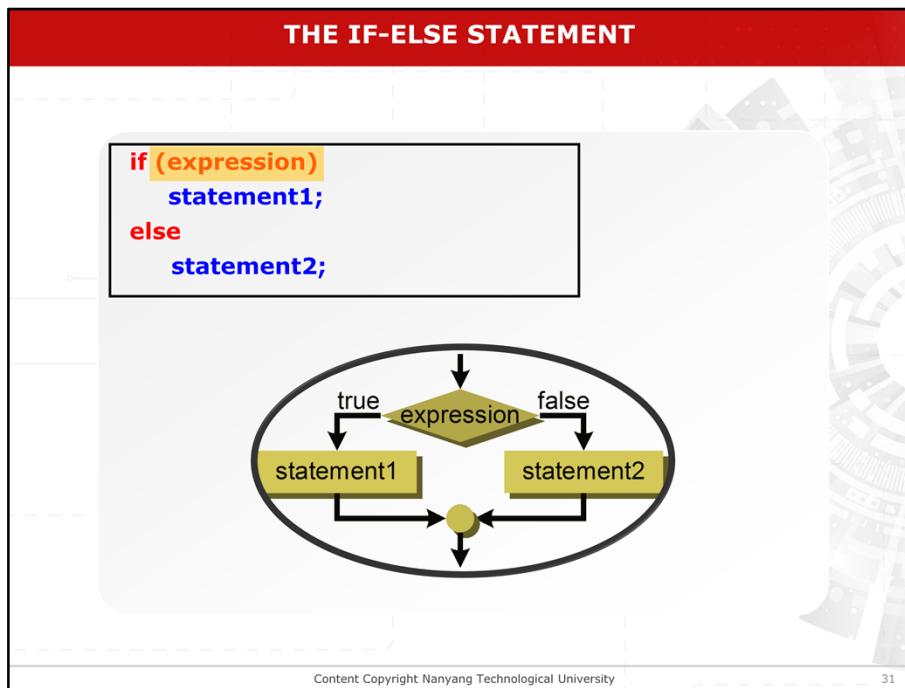
where **if** and **else** are reserved keywords..

The slide has a red header bar with the title "THE IF-ELSE STATEMENT". Below the title is a code block containing the pseudocode for an if-else statement. The code is as follows:

```
if (expression)
    statement1;
else
    statement2;
```

At the bottom left of the slide, there is a small text that says "Content Copyright Nanyang Technological University". At the bottom right, there is a small number "30".

When the **if-else** statement is executed, the **expression** is evaluated.



If **expression** is true, then **statement1** is executed and the control is passed to the program statement following the **if** statement. If **expression** is false, then **statement2** is executed. Both **statement1** and **statement2** may be a single statement terminated by a semicolon or a compound statement enclosed by {}.

THE IF-ELSE STATEMENT

```
/* This program determines the maximum value of num1 and num2 */
#include <stdio.h>
int main()
{
    int num1, num2, max;
    printf("Please enter two integers:");
    scanf("%d %d", &num1, &num2);
    /* write if-else code here */
    printf("The maximum of the \
two is %d\n",max);
    return 0;
}
```

Content Copyright Nanyang Technological University

32

In the example program, the purpose is to read in two integer numbers and determines the maximum number between them.

THE IF-ELSE STATEMENT: EXAMPLE

```
/* This program determines the maximum value of
num1 and num2 */
#include <stdio.h>
int main()
{
    int num1, num2, max;
    printf("Please enter two integers:");
    scanf("%d %d", &num1, &num2);
    if (num1 > num2)
        max = num1;
    else
        max = num2;
    printf("The maximum of the \
two is %d\n", max);
    return 0;
}
```

Content Copyright Nanyang Technological University 33

The program computes the maximum number of two input integers. The two input integers are read in and stored in the variables **num1** and **num2**.

THE IF-ELSE STATEMENT: EXAMPLE

```
/* This program determines the maximum value of
num1 and num2 */
#include <stdio.h>
int main()
{
    int num1, num2, max;
    printf("Please enter two integers:");
    scanf("%d %d", &num1, &num2);
    if (num1 > num2)
        max = num1;
    else
        max = num2;
    printf("The maximum of the \
two is %d\n",max);
    return 0;
}
```

Output
Please enter two integers: **9 4**

Content Copyright Nanyang Technological University 34

The program computes the maximum number of two input integers. The two input integers are read in and stored in the variables **num1** and **num2**.

THE IF-ELSE STATEMENT: EXAMPLE

```

/* This program determines the maximum value of
num1 and num2 */
#include <stdio.h>
int main()
{
    int num1, num2, max;
    printf("Please enter two integers:");
    scanf("%d %d", &num1, &num2);
    if (num1 > num2)
        max = num1;
    else
        max = num2;
    printf("The maximum of the \
two is %d\n",max);
    return 0;
}

```

Output

Please enter two integers: **9 4**

Content Copyright Nanyang Technological University

35

The **if** statement is then used to compare the two variables. If **num1** is greater than **num2**, then the variable **max** is assigned with the value of **num1**. Otherwise, **max** is assigned with the value of **num2**. The program then prints the maximum number through the variable **max**.

THE IF-ELSE STATEMENT: EXAMPLE

```
/* This program determines the maximum value of
num1 and num2 */
#include <stdio.h>
int main()
{
    int num1, num2, max;
    printf("Please enter two integers:");
    scanf("%d %d", &num1, &num2);
    if (num1 > num2)
        max = num1;
    else
        max = num2;
    printf("The maximum of the \
two is %d\n",max);
    return 0;
}
```

Output
Please enter two integers: 9 4
The maximum of the two is 9

Content Copyright Nanyang Technological University 36

If **num1** is greater than **num2**, then the variable **max** is assigned with the value of **num1**.

THE IF-ELSE STATEMENT: EXAMPLE

```
/* This program determines the maximum value of
num1 and num2 */
#include <stdio.h>
int main()
{
    int num1, num2, max;
    printf("Please enter two integers:");
    scanf("%d %d", &num1, &num2);
    if (num1 > num2)
        max = num1;
    else
        max = num2;
    printf("The maximum of the \
two is %d\n",max);
    return 0;
}
```

Output
Please enter two integers: -2 0

Content Copyright Nanyang Technological University 37

If **num1** is greater than **num2**, then the variable **max** is assigned with the value of **num1**.

THE IF-ELSE STATEMENT: EXAMPLE

```
/* This program determines the maximum value of
num1 and num2 */
#include <stdio.h>
int main()
{
    int num1, num2, max;
    printf("Please enter two integers:");
    scanf("%d %d", &num1, &num2);
    if (num1 > num2)
        max = num1;
    else
        max = num2;
    printf("The maximum of the \
two is %d\n",max);
    return 0;
}
```

Output
Please enter two integers: -2 0
The maximum of the two is 0

Content Copyright Nanyang Technological University

38

Otherwise, **max** is assigned with the value of **num2**. The program then prints the maximum number through the variable **max**.

THE IF-ELSE STATEMENT: EXAMPLE

```
/* This program determines the maximum value of
num1 and num2 */
#include <stdio.h>
int main()
{
    int num1, num2, max;
    printf("Please enter two integers:");
    scanf("%d %d", &num1, &num2);
    if (num1 > num2)
        max = num1;
    else
        max = num2;
    printf("The maximum of the \
two is %d\n",max);
    return 0;
}
```

Output
Please enter two integers: -2 0
The maximum of the two is 0

Content Copyright Nanyang Technological University

39

The program then prints the maximum number through the variable **max**.

The slide has a red header bar with the title "THE IF-ELSE IF-ELSE STATEMENT". Below the title is a code block containing:
`if (expression1)
 statement1;
else if (expression2)
 statement2;
else
 statement3;`

Content Copyright Nanyang Technological University 40

The if-else-if-else Statement

The format for **if-else if-else** statement is as shown.

The slide has a red header bar with the title "THE IF-ELSE IF-ELSE STATEMENT". Below the title is a code block:

```
if (expression1)
    statement1;
else if (expression2)
    statement2;
else
    statement3;
```

Content Copyright Nanyang Technological University

41

each of the **statement1**

THE IF-ELSE IF-ELSE STATEMENT

```
if (expression1)
    statement1;
else if (expression2)
    statement2;
else
    statement3;
```

Content Copyright Nanyang Technological University 42

statement2

The slide has a red header bar with the text "THE IF-ELSE IF-ELSE STATEMENT". The main content area contains a code snippet:

```
if (expression1)
    statement1;
else if (expression2)
    statement2;
else
    statement3;
```

At the bottom left of the slide, there is a small watermark-like image of a person sitting at a desk with a computer monitor. At the bottom center, the text "Content Copyright Nanyang Technological University" is visible, and at the bottom right, the number "43" is present.

And **statement3** can either be a single statement terminated by a semicolon or a compound statement enclosed by {}.

THE IF-ELSE IF-ELSE STATEMENT

```
if (expression1)
    statement1;
else if (expression2)
    statement2;
else
    statement3;
```

```

graph TD
    A{expression1} -- true --> B[statement1]
    B --> C(( ))
    A -- false --> D{expression2}
    D -- true --> E[statement2]
    E --> F(( ))
    D -- false --> G[statement3]
    G --> H(( ))
    C --> H
    F --> H
    
```

Content Copyright Nanyang Technological University

44

The if-else-if-else Statement

We may have as many **else if** parts as possible in the **if** statement provided it is within the compiler limit, where **statement1** is executed when the first **I-1** expressions are false and the **expression1** is true. If all the statements are false, then **statementN** will be executed. In any case, only one statement will be executed, and the rest will be skipped. The last **else** part is optional and can be omitted. If the last **else** part is omitted, then no statement will be executed if all the expressions are evaluated to be false.

THE IF-ELSE IF-ELSE STATEMENT

```
/* Program: Temperature reading. */
#include <stdio.h>
int main()
{
    float temp;
    printf("Temperature reading: ");
    scanf("%f",&temp);
    /* write if-else if-else code here */

    return 0;
}
```

Content Copyright Nanyang Technological University 45

In the example program, the purpose is to read in a value on temperature and determines whether the temperature is ok, too low or too high.

THE IF-ELSE IF-ELSE STATEMENT

```
/* Program: Temperature reading. */
#include <stdio.h>
int main()
{
    float temp;
    printf("Temperature reading: ");
    scanf("%f",&temp);
    /* write if-else if-else code here */

    if ((temp >= 100.0) && ( temp <= 120.0))
        printf("Temperature OK.\n");
    else if (temp < 100.0)
        printf("Temperature too low.\n");
    else
        printf("Temperature too high.\n");

    return 0;
}
```

Output
Temperature reading: **105.0**

Content Copyright Nanyang Technological University

46

The program first reads in the temperature input. If the input value is between 100 and 120,

THE IF-ELSE IF-ELSE STATEMENT

```
/* Program: Temperature reading. */
#include <stdio.h>
int main()
{
    float temp;
    printf("Temperature reading: ");
    scanf("%f",&temp);
    /* write if-else if-else code here */

    if ((temp >= 100.0) && ( temp <= 120.0))
        printf("Temperature OK.\n");
    else if (temp < 100.0)
        printf("Temperature too low.\n");
    else
        printf("Temperature too high.\n");

    return 0;
}
```

Output
Temperature reading: **105.0**
Temperature OK.

Content Copyright Nanyang Technological University 47

the string “Temperature OK.” will be printed the screen,

THE IF-ELSE IF-ELSE STATEMENT

```
/* Program: Temperature reading. */
#include <stdio.h>
int main()
{
    float temp;
    printf("Temperature reading: ");
    scanf("%f",&temp);
    /* write if-else if-else code here */

    if ((temp >= 100.0) && ( temp <= 120.0))
        printf("Temperature OK.\n");
    else if (temp < 100.0)
        printf("Temperature too low.\n");
    else
        printf("Temperature too high.\n");

    return 0;
}
```

Output
Temperature reading: **90**

Content Copyright Nanyang Technological University 48

else if the input value is less than 100,

THE IF-ELSE IF-ELSE STATEMENT

```
/* Program: Temperature reading. */
#include <stdio.h>
int main()
{
    float temp;
    printf("Temperature reading: ");
    scanf("%f",&temp);
    /* write if-else if-else code here */

    if ((temp >= 100.0) && ( temp <= 120.0))
        printf("Temperature OK.\n");
    else if (temp < 100.0)
        printf("Temperature too low.\n");
    else
        printf("Temperature too high.\n");

    return 0;
}
```

Output
Temperature reading: **90**
Temperature too low.

Content Copyright Nanyang Technological University 49

the string “Temperature too low” will be displayed

THE IF-ELSE IF-ELSE STATEMENT

```
/* Program: Temperature reading. */
#include <stdio.h>
int main()
{
    float temp;
    printf("Temperature reading: ");
    scanf("%f",&temp);
    /* write if-else if-else code here */

    if ((temp >= 100.0) && ( temp <= 120.0))
        printf("Temperature OK.\n");
    else if (temp < 100.0)
        printf("Temperature too low.\n");
    else
        printf("Temperature too high.\n");

    return 0;
}
```

Output
Temperature reading: **130**
Temperature too high.

Content Copyright Nanyang Technological University 50

Else temperature too high is printed on the screen.

SUMMARY

- After this lesson, you should be able to:
- Explain how if statement works with example
- Explain how if-else statement works with example
- Explain how if-else if-else statement works with example

Content Copyright Nanyang Technological University

51

In summary, after viewing this video lesson, your should be able to explain if-statement, if-else and if-else if-else statement with examples.