



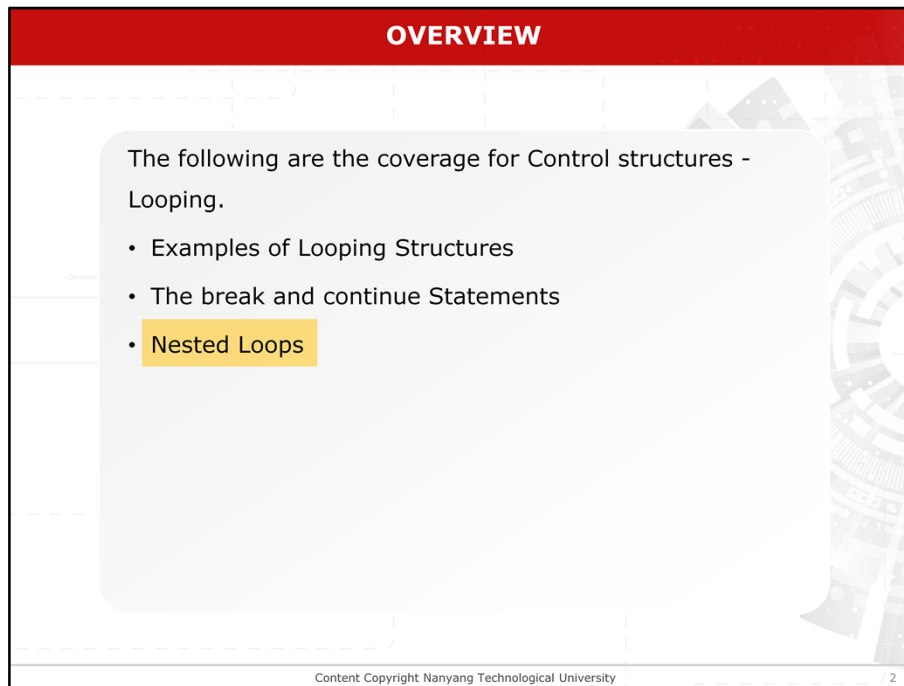
**NANYANG
TECHNOLOGICAL
UNIVERSITY**

CE1007/ CZ1007 DATA STRUCTURES

Lesson 3.3 Nested Loops

Assoc Prof Hui Siu Cheung

College of Engineering
School of Computer Science and Engineering



The slide features a red header with the word "OVERVIEW" in white. The main content is in a light gray box with a subtle background pattern of dashed lines and a gear-like graphic on the right. The text inside the box lists the topics to be covered, with "Nested Loops" highlighted in a yellow box. The footer contains the copyright notice and the page number.

OVERVIEW

The following are the coverage for Control structures - Looping.

- Examples of Looping Structures
- The break and continue Statements
- Nested Loops

Content Copyright Nanyang Technological University 2

Basic C Programming

There are 3 main sections to cover for Control structures (looping).



Learning objectives

LEARNING OBJECTIVES

At this lesson, you should be able to:

Content Copyright Nanyang Technological University 4

The slide features a red header with the text 'LEARNING OBJECTIVES'. Below the header is a large, light gray rectangular box with rounded corners, intended for listing learning objectives. The text 'At this lesson, you should be able to:' is positioned at the top left of this box. The background of the slide includes a faint, stylized graphic of a building and a gear. At the bottom, a thin black line separates the content from the footer, which contains the text 'Content Copyright Nanyang Technological University' and the number '4'.

At this lesson, you should be able to:

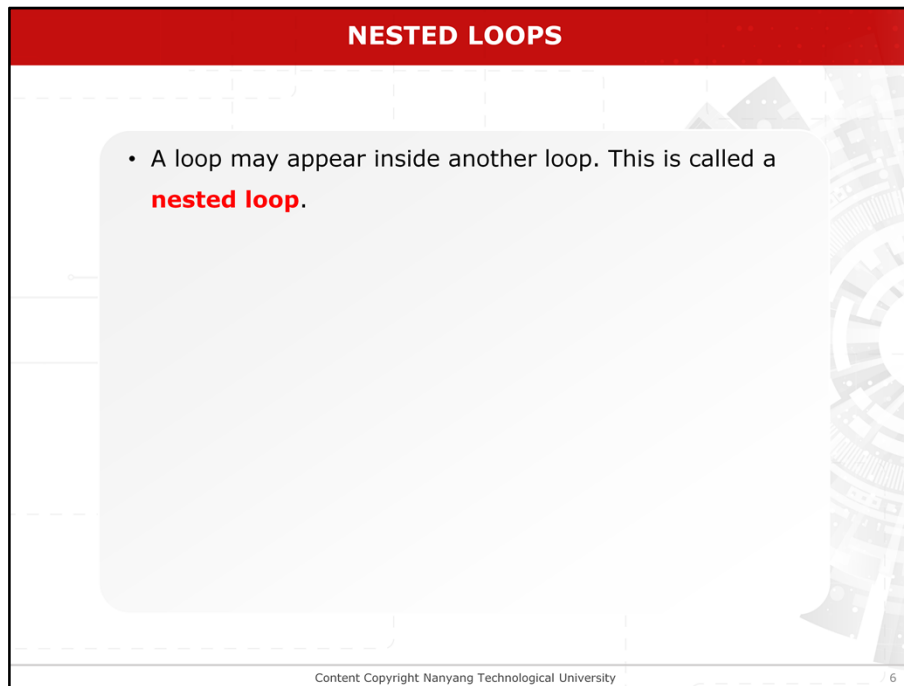
LEARNING OBJECTIVES

At this lesson, you should be able to:

- Execute the program using nested loops

Content Copyright Nanyang Technological University 5

At this lesson, you should be able to:



NESTED LOOPS

- A loop may appear inside another loop. This is called a **nested loop**.

Content Copyright Nanyang Technological University 6

Nested Loops

A loop may appear inside another loop. This is called a *nested* loop.

NESTED LOOPS

- A loop may appear inside another loop. This is called a **nested loop**.
- We can nest as **many levels** of loops as the hardware allows and we can nest **different types** of loops.

Content Copyright Nanyang Technological University 7

We can nest as many levels of loops as the system allows. We can also nest different types of loops.

NESTED LOOPS: EXAMPLE

```
/* count the number of different strings of a, b, c */
#include <stdio.h>
int main()
{
    char i, j;          /* for loop counters */
    int num = 0;         /* Overall loop counter */

    for (i = 'a'; i <= 'c'; i++) {
        for (j = 'a'; j <= 'c'; j++) {
            num++;
            printf("%c%c ", i, j);
        }
        printf("\n");
    }

    printf("%d different strings of letters.\n", num);
    return 0;
}
```

Click to play

Content Copyright Nanyang Technological University

8

Nested Loops: Example

In the program, it generates the different strings of letters from the characters 'a', 'b' and 'c'. The program contains a nested loop, in which one **for** loop is nested inside another **for** loop.

NESTED LOOPS: EXAMPLE

```

/* count the number of different strings of a, b, c */
#include <stdio.h>
int main()
{
    int space, asterisk, height, lines;
    printf("Please enter the height of the pattern:");
    scanf("%d", &height);
    for (lines = 1; lines <= height; lines++) {
        for (space = 1; space <= (height - lines); space++)
            putchar(' ');
        for (asterisk = 1; asterisk <= (2 * lines - 1); asterisk++)
            putchar('*');
        putchar('\n');
    }
    return 0;
}

```

Content Copyright Nanyang Technological University

Another example that uses nested loop is given in the following program. A sample input and output of the program is given here.

The program prints a triangular pattern according to the height entered by the user.

PROBLEM: IF-ELSE TO SWITCH

Mark	Grade
80 <= mark	A
70 <= mark < 80	B
60 <= mark < 70	C
50 <= mark < 60	D
40 <= mark < 50	E
mark < 40	F

```

int mark; char grade;
....
if (mark >= 80)
    grade = 'A';
else if (mark >= 70)
    grade = 'B';
else if (mark >= 60)
    grade = 'C';
else if (mark >= 50)
    grade = 'D';
else if (mark >= 40)
    grade = 'E';
else
    grade = 'F';
  
```

Content Copyright Nanyang Technological University 10

The purpose of this program is to convert the **if-else-if-else** statement into a **switch** statement in the mark-to-grade conversion problem. The **if-else-if-else** statement is used to control the grade to be assigned to the variable **grade** according to the input value on **mark**.

PROBLEM: IF-ELSE TO SWITCH

Mark	Grade
80 <= mark	A
70 <= mark < 80	B
60 <= mark < 70	C
50 <= mark < 60	D
40 <= mark < 50	E
mark < 40	F

Q: How to use the switch statement?

```

int mark; char grade;
switch ( ??? ) {
    case 10: ??
    case 9: ??
    case 8: ??
    case 7: ??
    case 6: ??
    case 5: ??
    case 4: ??
    default: ??
}

```

```

int mark; char grade;
....
if (mark >= 80)
    grade = 'A';
else if (mark >= 70)
    grade = 'B';
else if (mark >= 60)
    grade = 'C';
else if (mark >= 50)
    grade = 'D';
else if (mark >= 40)
    grade = 'E';
else
    grade = 'F';

```

Content Copyright Nanyang Technological University

11

How to use the switch statement?

PROGRAMMING SOLUTION: USING SWITCH

```
#include <stdio.h>
int main()
{
    int studentNumber = 0, mark;  char grade;
    printf("Enter StudentID: ");
    scanf("%d", &studentNumber);
    while (studentNumber != -1)
    {
        printf("Enter Mark: ");
        scanf("%d", &mark);
        if (mark >= 80)
            grade = 'A';
        else if (mark >= 70)
            grade = 'B';
        else if (mark >= 60)
            grade = 'C';
        else if (mark >= 50)
            grade = 'D';
        else if (mark >= 40)
            grade = 'E';
        else grade = 'F';
        printf("Grade = %c\n", grade);
        printf("Enter StudentID: ");
        scanf("%d", &studentNumber);
    }
    printf("Program terminating ...\n");
    return 0;
}
```

Click to play

SUMMARY

At this lesson, you should be able to:

- Execute the program using nested loops

Content Copyright Nanyang Technological University 13

In summary, after viewing this video lesson, you should be able to execute the program using nested loops.