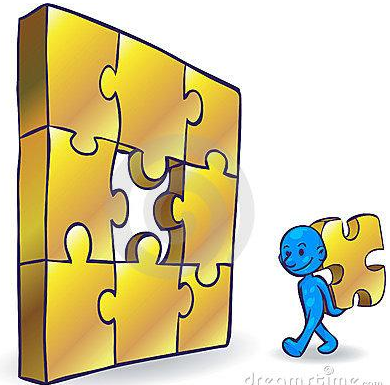# Topic 4

ALGORITHM DESIGN FOR SELECTION CONTROL STRUCTURE

Course Outline

- Analysis of Problems Requiring Selection Structure

- Boolean Values, Relational Operators, and Expressions

- Logical Operators

- Operator Precedence

- Algorithm Development for Selection Control Structure (Pseudo-code and Flowchart)

Selection Analogy

Making Decision:

You need to choose to make hot milo OR cold milo???

Selection Analogy

If it is raining…
I will use umbrella so that I don't get wet.

If it is raining…
- I will take my umbrella (If the condition is true)
- I will not take my umbrella (If the condition is not true)

Analysis of Problems Requiring Selection Structure

- To determine whether any positive number given is an even number

- To determine whether any positive number given is either an even number or an odd number.

- To determine whether any positive number entered is:
  - divisible by 2, or
 - divisible by 3, or
  - divisible by 5.

Operators

- **Operators** are the data connectors within expression and equations.
- They tell the computer:
  - Ways to process data.
  - Types of processing needs to be done.
- **THREE** types of **operators** used in calculation and problem solving include:
  1. Mathematical
  2. Relational
  3. Logical

Operators

- **Operand & Resultant**
- Data that connects and processes by the operator is called the **operands**.
- Meanwhile, when the operation is completed the answer that results is known as **Resultant**.
- Data type of operand and the resultant depend on the operator.

  – **Example:**

  operator

  5 + 7 = 12   resultant

  5 and 7 are operands
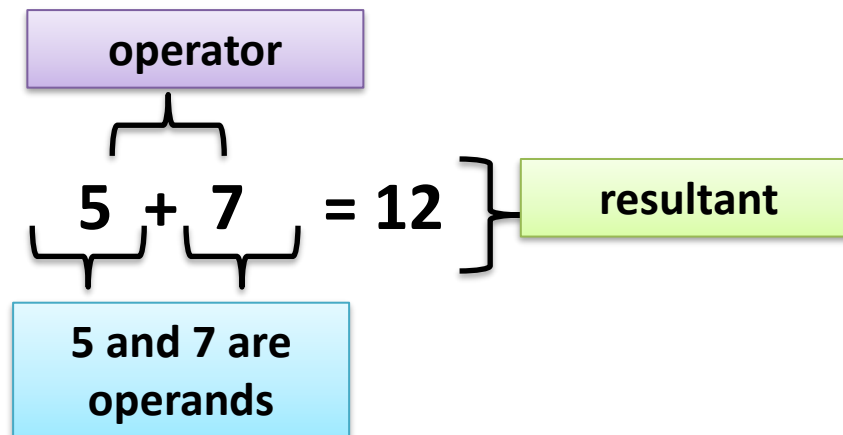
Operators

- **Mathematical Operators**

Action	Operator	Example	
		Operation	Resultant
Addition	+	3 + 5	8
Subtraction	-	7 - 4	3
Multiplication	*	8 * 5	40
Division	/	9 / 4	2
Modulus division	%	9 % 4	1

Operators

- **Relational Operators**

Operator	Computer Symbol	Example operation	Example resultant
Equal to	==	5 == 7	False
Less than	<	5 < 7	True
Greater than	>	5 > 7	False
Less than or equal to	<=	5<=7	True
Greater than or equal to	>=	5>=7	False
Not equal to	<>	5<>7	True

Operators

- **Relational Operators**
  - **Example**:
    - When a credit card customer's balance is less than RM500 (True), then the customer can charge another purchase. When the balance is not less than RM500 (False), then he cannot charge another purchase.
    - Expression should be :

operator

Balance < 500 (TRUE)

resultant

Balance and 500 are operands
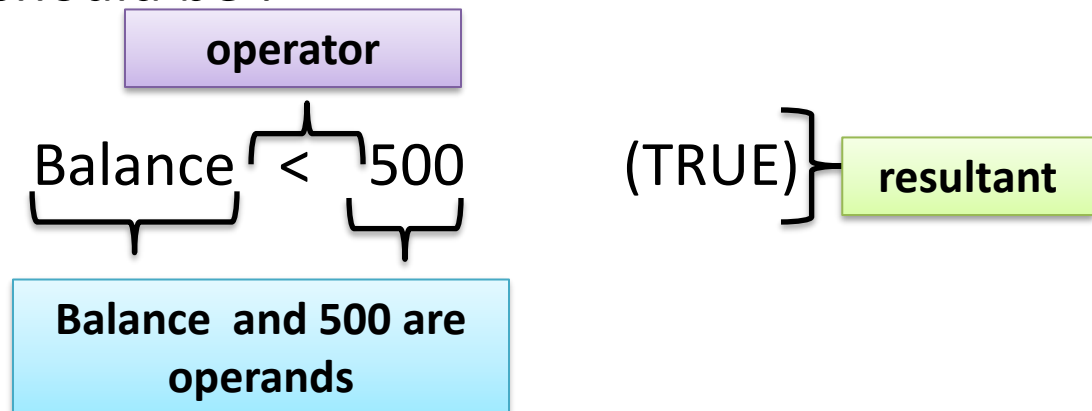
Operators

- **Logical Operators**

Operator	Description	Example operation	Example resultant
!	NOT	NOT true	False
&&	AND	True AND True	True
||	OR	True OR False	True

Boolean Expression

- Every decision a computer program makes involves evaluating a <u>Boolean expression.</u>

    – Boolean Expression: An expression whose value can be only true (1) or false (0).

Simple Boolean Expression

- Two numbers (operands) are <u>compared using a single relational operator.</u>

- Each produce Boolean expression (true or false result)

Expression	Meaning	Value
8 < 15	8 is less than 15	True
6 != 6	6 is not equal to 6	False
2.5 > 5.8	2.5 is greater than 5.8	False
5.9 <= 7.5	5.9 is less than or equal to 7.5	True

Compound Boolean Expression

- The Boolean operator **&&** (meaning **AND**) refers to the mathematical term <u>conjunction</u>.

- The Boolean operator **||** (meaning **OR**) refers to the mathematical term <u>disjunction</u>.

- The Boolean operator **!** (meaning **NOT**) is sometimes referred to as <u>negation</u>.
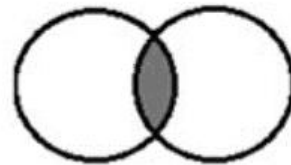
Compound Boolean Expression

Conjunction vs Disjunction
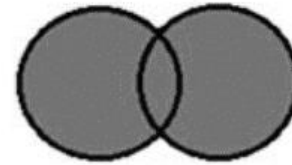
Case 1:

— age > 5 AND age ≤ 10

— age > 5 OR age ≤ 10

Case 2:

— age < 5 AND age ≥ 10

— age < 5 OR age ≥ 10

Conjunction

Disjunction
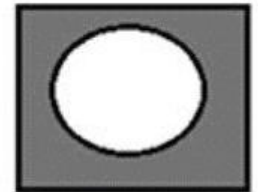
Negation

Boolean Expression and Boolean Value

- ! (NOT) Operator

  - Expression:

Expression	!(Expression)
true (1)	false (0)
false (0)	true (1)

- Example:

Expression	Value	Explanation
! ('A' > 'B')	true	Because 'A' > 'B' is **false**, ! ('A' > 'B') is **true**
! (6 <= 7)	false	Because 6 <= 7 is **true**, ! (6 <= 7) is **false**

Boolean Expression and Boolean Value

- && (AND) Operator

  - Expression:

Expression1	Expression2	Expression1 && Expression2
true (1)	true (1)	true (1)
true (1)	false (0)	false (0)
false (0)	true (1)	false (0)
false (0)	false (0)	false (0)

Boolean Expression and Boolean Value

- Example:

Expression	Value	Explanation
(14 > = 5) && ('A' < 'B')	true	Because (14 > = 5) is **true**, ('A' < 'B') is **true**, and **true** && **true** is **true**, the expression evaluates to **true**.
(24 >= 35) && ('A' < 'B')	false	Because (24 >= 35) is **false**, ('A' < 'B') is **true**, and **false** && **true** is **false**, the expression evaluates to **false**.

Boolean Expression and Boolean Value

- || (OR) Operator

  - Expression:

Expression1	Expression2	Expression1 || Expression2
true (1)	true (1)	true (1)
true (1)	false (0)	true (1)
false (0)	true (1)	true (1)
false (0)	false (0)	false (0)

Boolean Expression and Boolean Value

• Example:

Expression	Value	Explanation
(14 > = 5) || ('A' > 'B')	true	Because (14 > = 5) is **true**, ('A' > 'B') is **false**, and **true** || **false** is **true**, the expression evaluates to **true**.
(24 >= 35) || ('A' >'B')	false	Because (24 >= 35) is **false**, ('A' > 'B') is **false**, and **false** && **false** is **false**, the expression evaluates to **false**.
('A' <='a') || (7 !=7)	true	Because ('A' <='a') is **true**, (7 !=7) **false**, and **true** || **false** is **true**, the expression is evaluates to **true**.

Problem Example

Situation:
- An Identity Card or a Student Card is required for a student to sit for the final exam.
- When the student has an identity card, student can sit for the final exam.
- When the student has a Student Card, student can sit for the final exam.

Analysis:
- Expression is written as **Identity Card OR Student Card**
- When both operand is True, resultant is True
- When both operand is False, resultant is False
- When the operand True OR False, resultant is True

Conclusion: Therefore, student can sit for the final exam when he/she has either one card.

Problem Example

Situation:
- An Identity Card and a Student Card are required for a student to sit for the final exam.

Analysis:
➢ Expression is written as **Identity Card AND Student Card**
➢ When both operand are True, resultant is True
➢ When both operand are False, resultant is False
➢ When the is operand True AND False, resultant is False

Conclusion: Student can sit for the final exam only when he/she has both cards.

Precedence of Operator

- Expressions with higher-precedence operators are evaluated first

Operators	Precedence
++, --	First
!, +, - (unary operators)	Second
*, /, %	Third
+, - (binary operator)	Fourth
<, <=, >=, >	Fifth
==, !=	Sixth
&&	Seventh
||	Eighth
= (assignment operator)	Last

Precedence of Operator: Exercise

Example:

- Solve the following expression
  a) 2 / 3 * 4 ^ 2
  b) 10 + 6 / 4 – 14 % 3 * (7 – 3)
  c) (10 >= 10) && (20 < 20)
  d) 65 == 65 && 65 >= 70 || 90 != 105

Precedence of Operator: Exercise

- Solution
    - a) 3 - 2 * 4 ^ 2   **-29**
    - b) 10 + 6 / 4 – 14 % 3 * (7 – 3)   **3**
    - c) (10 >= 10) && (20 < 20)   **false**
    - d) 65 == 65 && 65 >= 70 || 90 != 105   **true**

SELECTION

Selection

- The selection structure <u>tests a condition</u>, then <u>executes one sequence of statements</u> instead of another, depending on whether the condition is true or false.

- A condition is any variable or expression that returns a Boolean value (TRUE or FALSE).

- The variations of selection structure are:
  - ➤ **One-way selection** (Single-alternative selection structure)
  - ➤ **Two-way selection (**Dual-alternative selection structure)
  - ➤ **Multiway selection**
    - ○ If-else-if
    - ○ Several if
  - ➤ **Nested selection**

Selection

- Selection structure, with the use of relational operators, can do:

  - **Numeric comparison**

    - Example: num1 > 10

  - **Character comparison**

    - Example: if (grade == 'A')

  - **String comparison**

    - Example: Month != "February"

ONE WAY SELECTION

One-Way Selection

- Syntax form:

```
if (condition)
    statement1
endIf
```

- Explanation:
  - The **condition** is evaluated first
  - If the **condition** is evaluated to TRUE, then **statement1** is executed

Flowchart
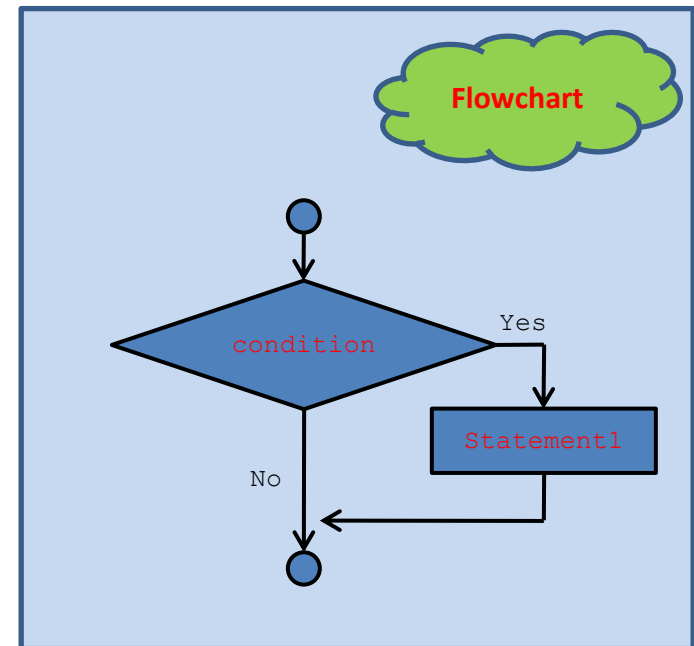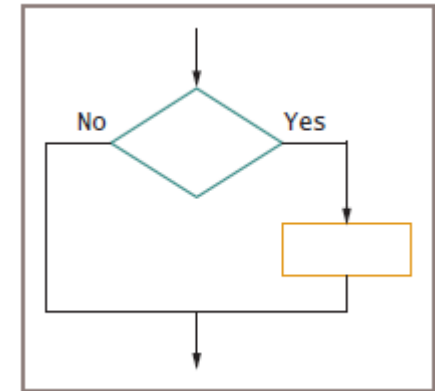
One-Way Selection: Example

- Example: Determine whether any positive number given is an even number.

- Pseudocode:

```
Start
    Read number
    if (number % 2 == 0)
        Show "Even"
    endIf
End
```


Flowchart

Start

Read number

number % 2 == 0

Yes

No

Show "Even"

End

Expected Output:
✓ If the input value for variable **number** is 6:
    ✓ Output displayed: Even

One-Way Selection: Exercise 1

➢Problem

To determine an invalid grade if the grade is out of range (0-100)

One-Way Selection: Exercise 1

- Pseudocode

```
START
     1. Declare grade

     2. Prompt "Please enter a grade: "
     3. Get grade

     4. Check grade
            IF grade < 0 OR grade > 100
                  Print " The grade is not valid"

     5. Print "Thank You!"
END
```

One-Way Selection: Exercise 1

Draw the flowchart for the previous pseudocode.

One-Way Selection: Exercise 2

➢ Problem

Write the complete pseudocode and flowchart that receive CGPA from user, and display academic status based on CGPA below.

CGPA	Status
More than 1.80	Passed

TWO WAY SELECTION

Two-Way Selection

- Syntax form:

```
if (condition)
    statement1
else
    statement2
endIf
```

- Explanation:
  - If the **condition** is evaluated to TRUE, then **statement1** is executed, and **statement2** is skipped.
  - If the **condition** is evaluated to FALSE, then **statement2** is executed, and **statement1** is skipped.
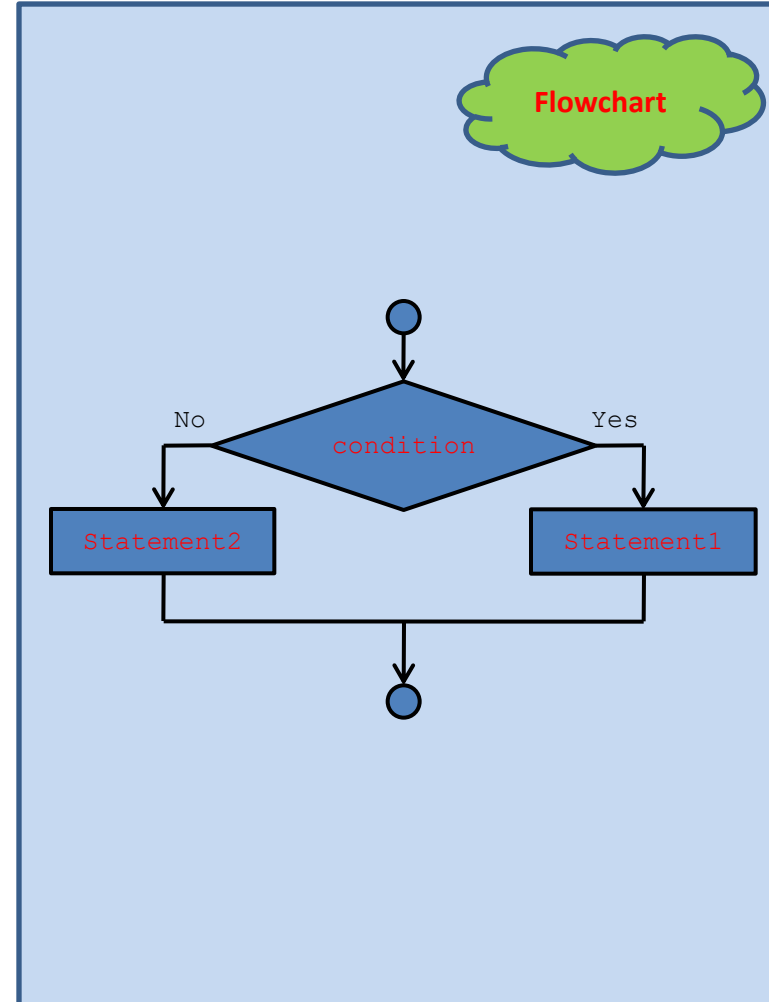
Two-Way Selection: Example

- Example: Determine whether any positive number given is either an even number or an odd number.

- Pseudocode:

```
Start
    Read number
    If (number % 2 == 0)
        Show "Even"
    Else
        Show "Odd"
    EndIf
End
```
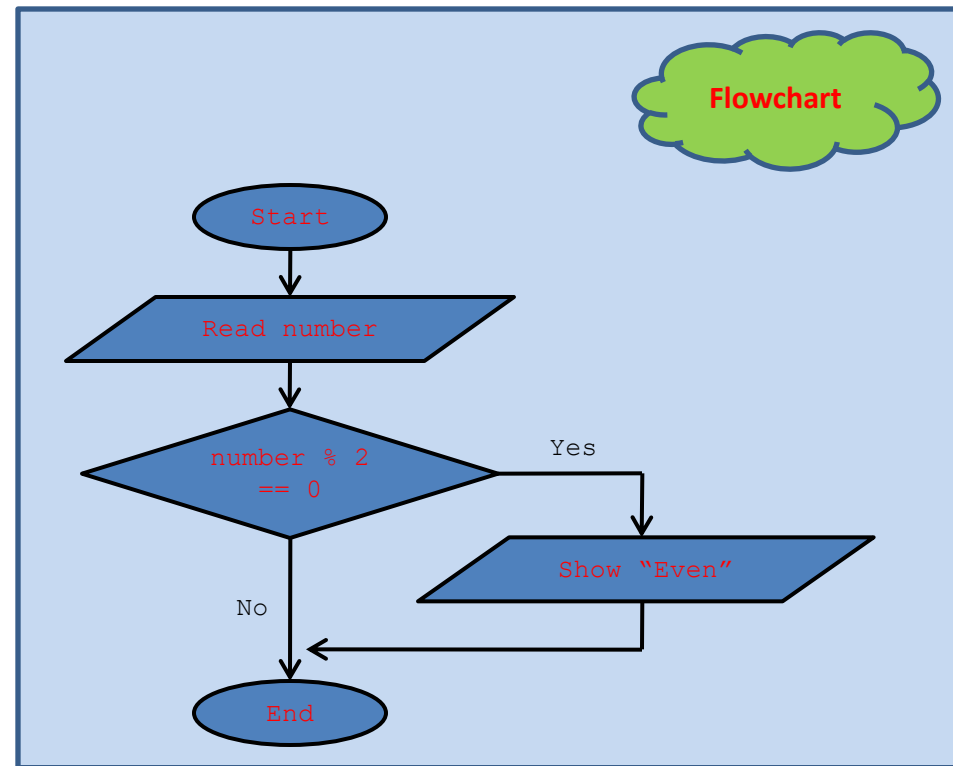

Flowchart

Start

Read number

number % 2 == 0

No   Yes

Show "Even"   Show "Odd"

End
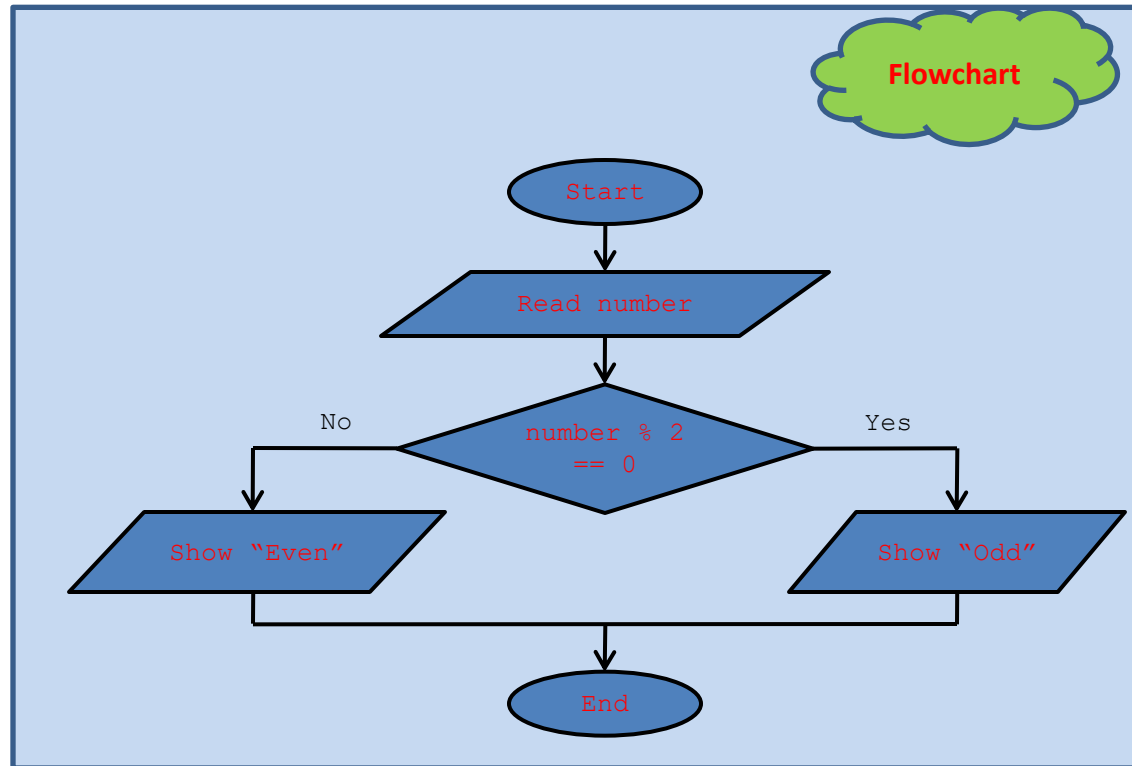
Expected Output:
- ✓ If the input value for variable **number** is 6:
  - ✓ Output displayed: Even
- ✓ If the input value for variable **number** is 7:
  - ✓ Output displayed: Odd

Two-Way Selection: Exercise 1

➤ Problem

To determine an invalid grade if the grade is out of range (0-100) and output the grade if the grade is within the range.

Two-Way Selection: Exercise 1

- Pseudocode

```
START
      1. Declare grade

      2. Prompt "Please enter a grade: "
      3. Get grade

      4. Check grade
            IF grade < 0 OR grade > 100
                    Print "The grade is not valid"
            ELSE
                    Print "The grade is" , grade

      5. Print "Thank You!"
END
```

Two-Way Selection: Exercise 1

Draw the flowchart for the previous pseudocode.

Two-Way Selection: Exercise 2

➢ Problem

Write the complete pseudocode and flowchart that receive CGPA from user, and display academic status based on CGPA below.

CGPA	Status
1.80 – 4.00	Passed
Otherwise	Failed

MULTI WAY SELECTION

Multi-Way Selection (if-else-if)

- Syntax form:

  ```
  if (condition1)
      statement1
  Else if (condition2)
      statement2
  Else if (condition3)
      statement3
  Else
      statement4
  EndIf
  ```

- Explanation:
  - If the **condition1** is evaluated to TRUE, then **statement1** is executed, and **statement2, statement3, statement4** are skipped.
  - **Statement2** will be executed if the **condition1** is evaluated to FALSE, and only if **condition2** is evaluated to TRUE.

Flowchart
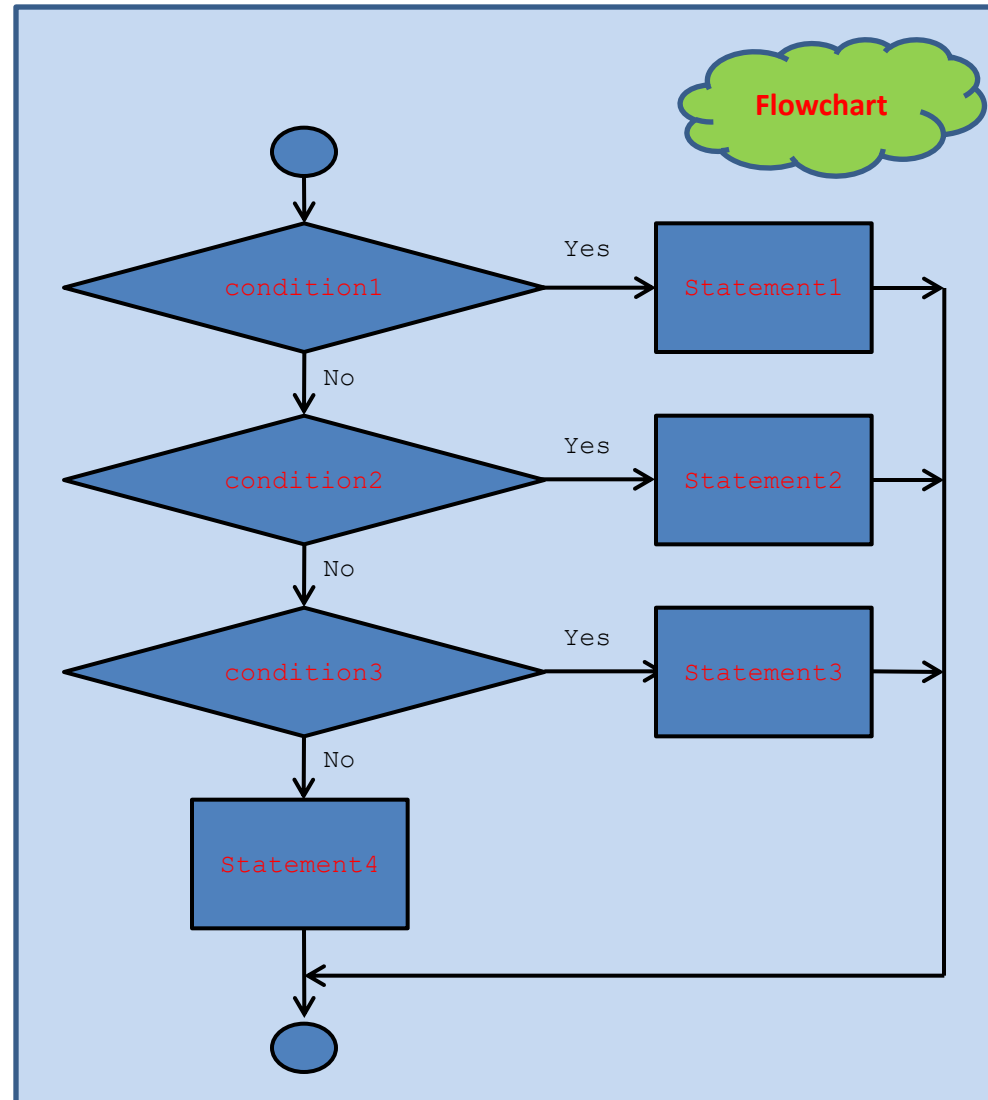
Multi-Way Selection (if-else-if): Example

Example: To determine whether any positive number entered is:
- ✓ divisible by 2, or
- ✓ divisible by 3, or
- ✓ divisible by 5.

```
Start
    Read number
    If (number % 2 == 0)
        Print "Divisible by 2", newline
    Else If (number % 3 == 0)
        Print "Divisible by 3", newline
    Else If (number % 5 == 0)
        Print "Divisible by 5", newline
    Else
        Print "NOT divisible by 2 or 3 or 5"
    EndIf
End
```

Expected Output, for the following input values:

1. Input value → 10	2. Input value → 6	3. Input value → 9
10 Divisible by 2	6 Divisible by 2	9 Divisible by 3

Flowchart

Start → Read number → number % 2 == 0 —Yes→ Print "Divisible by 2", newline

number % 3 == 0 —Yes→ Print "Divisible by 3", newline

number % 5 == 0 —Yes→ Print "Divisible by 5", newline

No → Print "NOT divisible by 2 or 3 or 5" → End
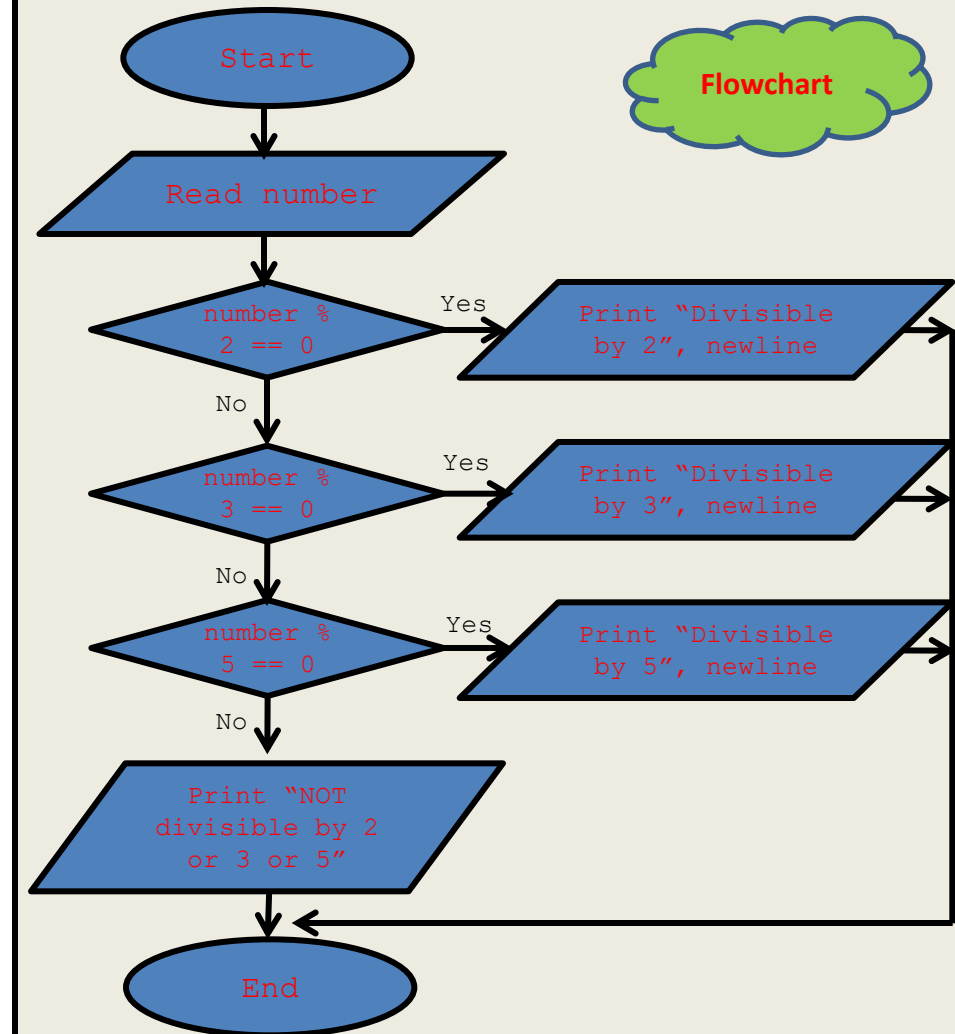
Multi-Way Selection (Several if)

- Syntax form:

```
if (condition1)
    statement1
endIf
if (condition2)
    statement2
endIf
if (condition3)
    statement3
endIf
if (condition4)
    statement4
endIf
```

- Explanation:
  - Any statement with a TRUE condition will be executed, statement with a FALSE condition will be skipped.
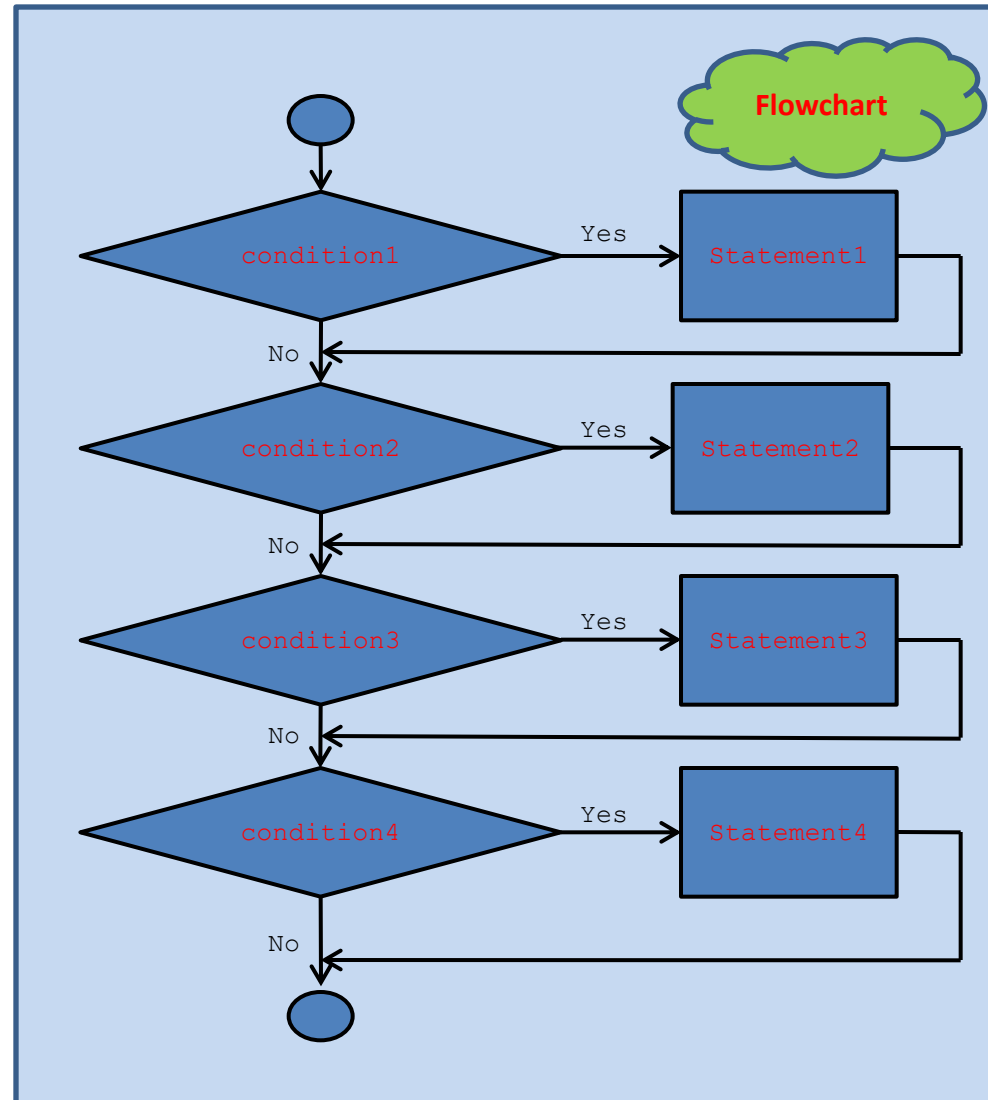
Multi-Way Selection (Several if): Example

Example: To determine whether any positive number entered is:
- ✓ divisible by 2, or
- ✓ divisible by 3, or
- ✓ divisible by 5.

```
Start
    Read number                    Pseudocode
    If (number % 2 == 0)
        Print "Divisible by 2", newline
    EndIf
    If (number % 3 == 0)
        Print "Divisible by 3", newline
    EndIf
    If (number % 5 == 0)
        Print "Divisible by 5", newline
    EndIf
End
```

Flowchart

Expected Output, for the following input values:

1. Input value → 10	2. Input value → 6	3. Input value → 9
10 **Divisible by 2** **Divisible by 5**	**6** **Divisible by 2** **Divisible by 3**	**9** **Divisible by 3**

Multi-Way Selection : Exercise

➢ Problem

Write the complete pseudocode and flowchart that receive CGPA from user, and display academic status based on CGPA below.

CGPA	Status
>= 3.50	Dean's List
1.80 – 3.49	Passed
< 1.80	Failed

NESTED SELECTION

Nested Selection (Nested if)

☐ Situations where action is depends on two or more different conditions.

• Syntax form:
```
if (condition1)
    if (condition2)
        statement1
    Else
        statement2
    endIf
Else if (condition3)
    if (condition4)
        statement3
    Else
        statement4
    endIf
endIf
```


Flowchart
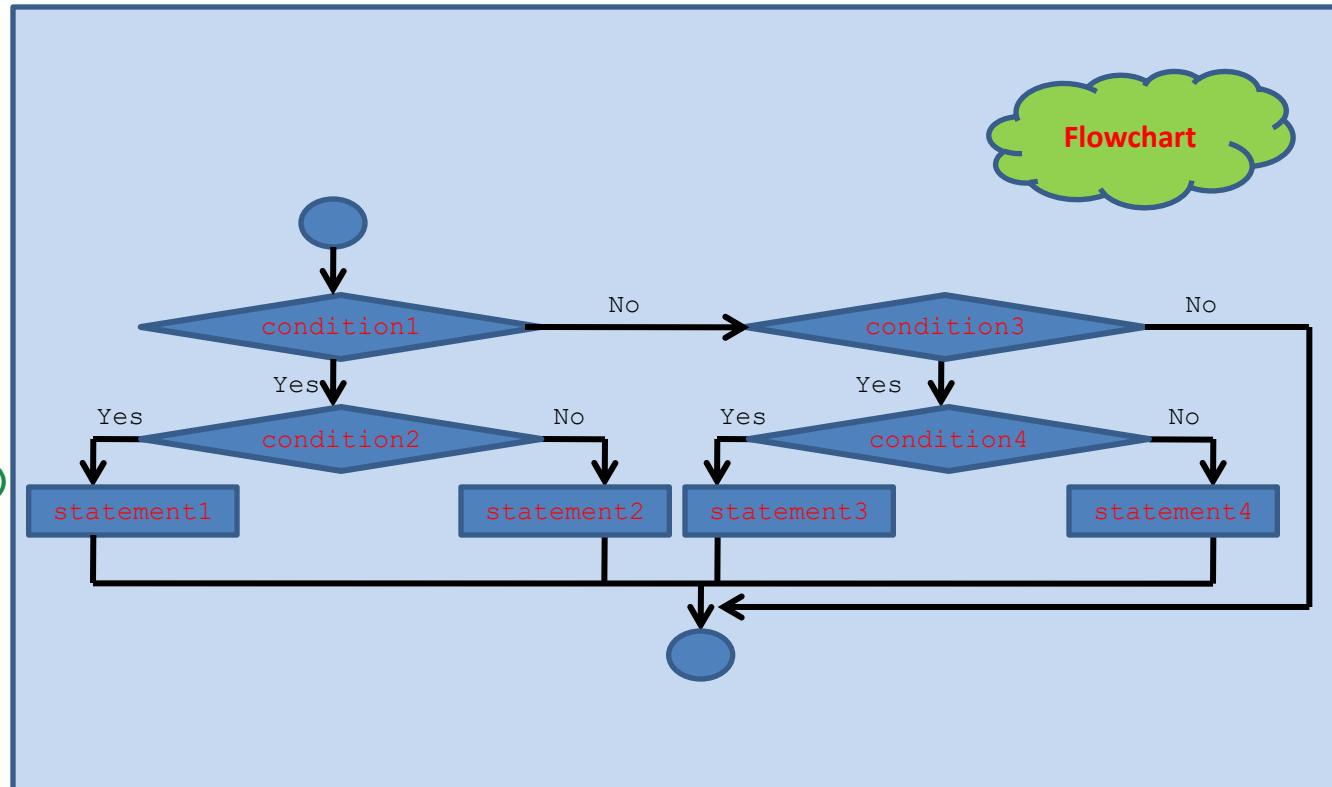
Nested Selection (Nested if): Example

Example: To calculate charges based on the services code and type of membership as shown in table.

SPA Services Code	Price (RM)	Discount	
		Type of Membership	
		Member (M)	Non-Member
Traditional Body Massage (B)	120.00	20%	10%
Foot Massage (T)	45.00	5%	-

Expected Output for the following input value:

✓ Service's code → T

✓ Membership → M

```
Enter Service's code (B/T): T
Enter membership (M-Member): M

Charges : RM42.75
```
OUTPUT

```
Start
    Display "Enter Service's code (B/T): "
    Read code
    Display "Enter membership (M-Member): "
    Read category

    if (code == 'B')
        price = 120.00
        if (category == 'M')
            disc = 0.2 * price
        else
            disc = 0.1 * price
        endif
    Else if (code == 'T')
        price = 45.00
        if (category == 'M')
            disc = 0.05 * price
        else
            disc = 0.0
        endif
    Endif
    charge = price - disc
    Display "Charges: RM", charge
End
```
PSEUDOCODE

Nested Selection (Nested if): Example

FLOWCHART

```
                              Start
                                │
                                ▼
                   Display "Enter Service's
                        code (B/T): "
                                │
                                ▼
                           Get code
                                │
                                ▼
                   Display "Enter Membership
                        (M-Member): "
                                │
                                ▼
                         Get category
                                │
                                ▼
                         code ==          No          code ==          No
                          'B'    ────────────►         'T'    ──────────────┐
                                │Yes                   │Yes                  │
                                ▼                       ▼                    │
                      price = 120.00            price = 45.00               │
                                │                       │                    │
                   Yes          ▼          No   Yes     ▼        No          │
               ┌──────  code ==  ──────┐  ┌──────  code ==  ──────┐         │
               │         'M'            │  │         'M'            │         │
               ▼                        ▼  ▼                        ▼         │
        disc = 0.2 *          disc = 0.1 *    disc = 0.05 *      disc = 0.0   │
          price                 price            price                        │
               └────────┐          │    │          └──────────┐              │
                        ▼          │    │                                    │
                  charge = price - disc                                      │
                                │                                            │
                                ▼                                            │
                   Display "Charges: RM", charge                            │
                                │ ◄──────────────────────────────────────────┘
                                ▼
                              End
```
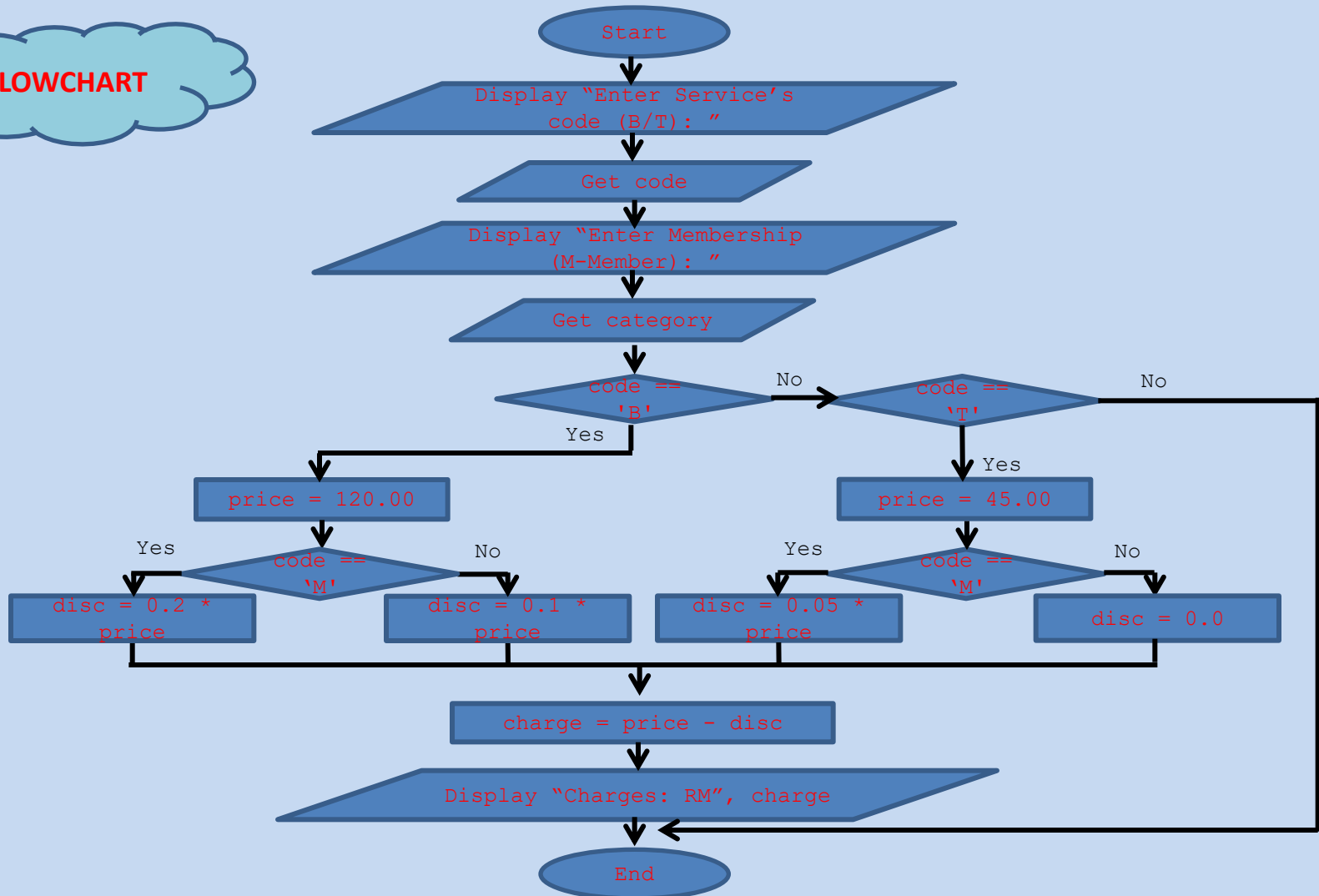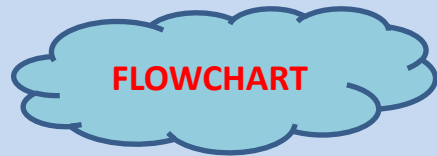
Nested Selection (Nested if): Exercise

➤ Problem

Find the total museum admission fee **based on nationality and age** as in the following table.

Total Fee = Fee x Quantity

Nationality	Age	Fee
1) Malaysian	< 60	10.00
	≥ 60	5.00
2) Non-Malaysian	< 60	25.00
	≥ 60	15.00

Summary

This topic explains

- The selection mechanism to solve a problem.

- There are three (3) operators in selection problem solving (Mathematical, Relational and Logical).

- Every selection involves evaluating a Boolean expression (true / false).

- The selection structure tests a condition, then executes one sequence of statements depending on whether the condition is true or false.

- The are variations of selection structure are:
  - One-way selection
  - Two-way selection
  - Multiway selection
  - Nested selection

References

– Farrell, Joyce, *Programming Logic and Design Comprehensive*, 2nd edition, Course Technology 2002.

– Liang, Y.D., *Introduction to Programming with C++*, 2nd edition, Pearson Higher Education, 2010.

– N Mohamad, M Puteh. *Problem Solving with C++*, UPENA, 2006.

– Sprankle, M., & Hubbard, J. (2009). Problem Solving & Programming Concepts (8th ed.). Upper Saddle River, NJ: Pearson Education.