The Branching

Contents of Lecture:

- Introduction
- Conditional structure
- Simple If
- If-else Statement
- Nested If Statement
- **Homework number (3) delivered next week**

Introduction

- A program is usually not limited to a linear sequence of instructions.
- During its process it may take decisions, for those purpose the branch structures that serve to specify what has to be done by our program.

- Branching statements allows us to redirect the flow of program execution. In other word branch statements that allow us to select and execute specific blocks of code while skipping other sections.

Conditional structure:

- In practice, the computer is presented not with a *true / false* statement, but with a question having a "Yes" or "No" answer using relational operators(>, <, >=, <=, =, ≠).
- For example: if A = 10, B = 20, K = 5 and SALES = 10000, then:

Condition (Question)	**"Answer"**
If (A = B)?	No
If (B > A)?	Yes
If (K <= 25)?	Yes
If (SALES >= $500.000)?	Yes

- With each question, the computer can be programmed to take a different course of action depending on the answer.

- A step in an algorithm that leads to more than one possible continuation is called a **decision**.

- In **flowcharting**, the diamond-shaped symbol is used to indicate a decision.
- The question is placed inside the symbol, and each alternative answer to the question is used to label the exit arrow which leads to the appropriate next step of the algorithm.

- The decision symbol is the only symbol that may have more than one exit.

Simple If

- ❖ If we have a code that we sometimes want to execute and sometimes we want to skip we can use if statement.
- ❖ Can use **if** keyword to execute a statement or block only if a condition is true.

- ❖ If this condition is *true*, statement is executed. If it is *false*, statement is ignored (not executed) and the program continues right after this conditional structure.

- ❖ **For example**, the following code out word "hundred" if the value stored in the *num_1* variable is equal 100:

If-else Statement

- The **if-else** keyword is used to execute statement1 only if a condition is *true*, and execute the statement2 if the condition is *false*, its form as following:

```
If (condition)
        Statement1
Else
        Statement2
EndIf
```

- **For example**, the following code output num_1 if it is greater than num_2 otherwise output num_2:

1. Pseudocode:

2. Flowchart

Nested If Statement:

- By using *nested if*, it is possible to combine several conditions.
- Only the statements following the first condition that is found to be true will be executed, all other statements will be skipped.
- The statements of the final *Else* will be executed if none of the conditions are true.

```
If (condition1)
    -- statements
elsIf (condition2)
-- more statements
elsIf (condition3)
    -- more statements
...
elseIf (condition n)
    -- statement n
Else
-- other statements
EndIf
```

- **Example for nested if**, the following pseudo code and the flow chart represents student's grade which depend on his degree:

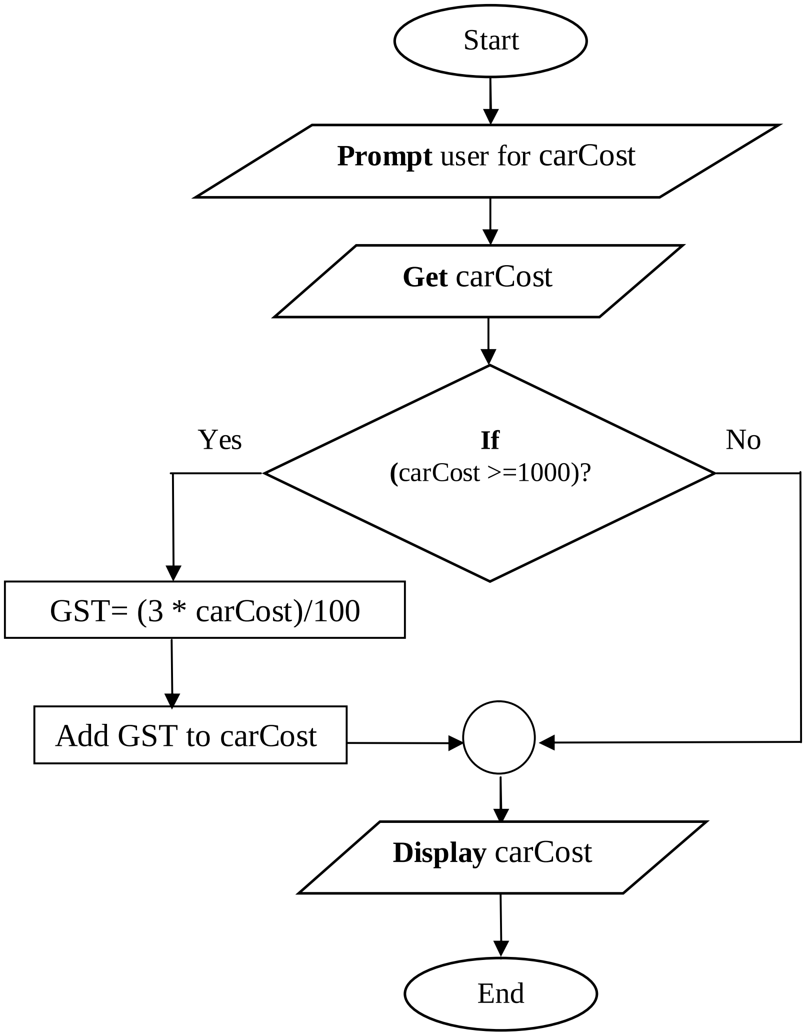
Homework number (3) delivered next week

A. Type an algorithm shows if the score variable is greater than 50 or not, and output the appropriate message (write pseudo code and a flowchart).

B. Convert the following pseudo code to a flowchart

```
1. Begin
2. Read cost
3. IF cost is >= $300 THEN
           Calculate commission =  cost * 1%
    else
           Calculate commission =  cost * 3%
    EndIf
4. Display cost , commission
5. End
```

C. Convert the following flowchart to pseudo code


Start
Prompt user for carCost
Get carCost
If
(carCost >=1000)?
Yes
No
GST= (3 * carCost)/100
Add GST to carCost
Display carCost
End