PROG0101 FUNDAMENTALS OF PROGRAMMING

Chapter 6
Branching
Branching

Topics

• Branching Statement
• IF Statement
• Switch Statement
Branching Statement

- A program consists of a number of statements which are usually executed in sequence.
- Programs can be much more powerful if we can control the order in which statements are run.
- The Branching Statements or Control Statements let the program makes a decision about what to do next.
IF Statement

• The **IF Statement** is a powerful decision making statement and is used to control the flow of execution of statements.
• It is basically a two-way decision statement and is used in conjunction with an expression.
IF Statement

- It allows the computer to evaluate the expression first and then, depending on whether the value of the expression (relation or condition) is 'true' (non-zero) or 'false' (zero), it transfers the control to a particular statement.
- This point of program has two paths to follow, one for the true condition and the other for the false condition.
- The if statement may be implemented in different forms depending on the complexity of the conditions to be tested.
IF Statement

• Two types of IF Statement:
  – Simple IF Statement
  – IF-ELSE Statement
Simple IF Statement

- The statement-block may be a single statement or a group of statements.
- If the test expression is true, the statement-block will be executed; otherwise the statement-block will be skipped and the execution will jump to statement-x.
- Remember, when the condition is true both the statement-block and the statement-x are executed in sequence.
Simple IF Statement
IF-ELSE Statement

• The if....else statement is an extension of the simple if statement.
• If the test expression is true , then the true-block statement(s), immediately following the if statement are executed; otherwise the false-block statement(s) are executed.
• In either case, either true-block or false-block will be executed, not both.
IF-ELSE Statement
Example 1

The flow chart shows a program which asks a user to enter a number, and then compare the number given either it is bigger or smaller than 5.
Example 2

The flow chart shows a program to check whether the student “Passed” or “Failed”.

*false*

Grade >= 60

*true*

print “Passed”

print “Failed”
SWITCH Statement

• In computer programming, a **Switch Statement** (also known as **Case Statement**) is a type of control statement that exists in most modern imperative programming languages.
• This is another form of the multi way decision.
• A switch statement is a selection statement that lets you transfer control to different statements within the switch body depending on the value of the switch expression.
SWITCH Statement

• If the value of the switch expression equals the value of one of the case expressions, the statements following that case expression are processed.
• If not, the default label statements, if any, are processed.
Example 1

```
Evaluate code

- code == 'A'?
  - Yes: discount = 0.0
  - No: code == 'B'?
    - Yes: discount = 0.1
    - No: code == 'C'?
      - Yes: discount = 0.2
      - No: Default: discount = 0.3
```
Example 2

- Operator is `+`?  
  - Yes: Print $a+b$, break;
  - No:
    - Operator is `-`?  
      - Yes: Print $a-b$, break;
      - No:
        - Operator is `*`?  
          - Yes: Print $a*b$, break;
          - No:
            - Operator is `/`?  
              - Yes: Print $a/b$, break;
              - No: Error, break;
        - Next statements following switch
