SELECTION STATEMENT (IF)

- IF statements with an ELSE clause are written as follows:

```
IF ChallengerScore > ChampionScore
  THEN
    IF ChallengerScore > HighestScore
      THEN
        OUTPUT ChallengerName, " is champion and highest scorer"
      ELSE
        OUTPUT Player1Name, " is the new champion"
    ENDIF
  ELSE
    OUTPUT ChampionName, " is still the champion"
    IF ChampionScore > HighestScore
      THEN
        OUTPUT ChampionName, " is also the highest scorer"
    ENDIF
ENDIF
```

SELECTION STATEMENT (CASE OF)

- CASE statements allow one out of several branches of code to be executed, depending on the value of a variable
- Unlike IF statement that has only two possible output (when it is true and false)
- This make code easier to read

```
INPUT Move
CASE OF Move
  'W' : Position ← Position - 10
  'E' : Position ← Position + 10
  'A' : Position ← Position - 1
  'D' : Position ← Position + 1
  OTHERWISE OUTPUT "Beep"
ENDCASE
```

ITERATION (COUNT-CONTROLLED LOOP)

- FOR loop is a count-controlled loop, it repeats for a number of times

```
Total ← 0
FOR Row ← 1 TO MaxRow
    RowTotal ← 0
    FOR Column ← 1 TO 10
        RowTotal ← RowTotal + Amount[Row, Column]
    NEXT Column
    OUTPUT "Total for Row ", Row, " is ", RowTotal
    Total ← Total + RowTotal
NEXT Row
OUTPUT "The grand total is ", Total
```

ITERATION (COUNT-CONTROLLED LOOP)

- FOR loop is a count-controlled loop, it repeats for a number of times

```
FOR <identifier> ← <value1> TO <value2> STEP <increment>
    <statements>
NEXT <identifier>
```

ITERATION (PRE-CONDITION LOOP)

- The condition is tested before the statements, and the statements will only be executed if the condition evaluates to TRUE
- After the statements have been executed the condition is tested again. The loop terminates when the condition evaluates to FALSE

```
WHILE Number > 9 DO
      Number ← Number - 9
ENDWHILE
```

ITERATION (POST-CONDITION LOOP)

- The statements in the loop will be executed at least once
- The condition is tested after the statements are executed and if it evaluates to TRUE the loop terminates, otherwise the statements are executed again

```
REPEAT
    OUTPUT "Please enter the password"
    INPUT Password
UNTIL Password = "Secret"
```

REPEAT UNTIL VS WHILE DO LOOP

- Write pseudocode using REPEAT UNTIL and WHILE DO LOOP to validate a number, the program allow positive numbers to be input only