

情報基礎B (Computer Literacy)

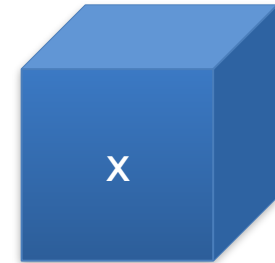
Lecture 7: If condition with VBA

Matias Korman

Tohoku University Graduate School of Information Sciences
System Information Sciences
Design and Analysis of Information Systems

Variables

- A box used to store information
- Must declare the type first
 - e.g. prepare a box “x” to store an integer
 - **Dim** x **As** Integer
 - Declare variable “x” to store integers
 - **Dim** name **As** String
 - Stores text instead



Numeric Data types

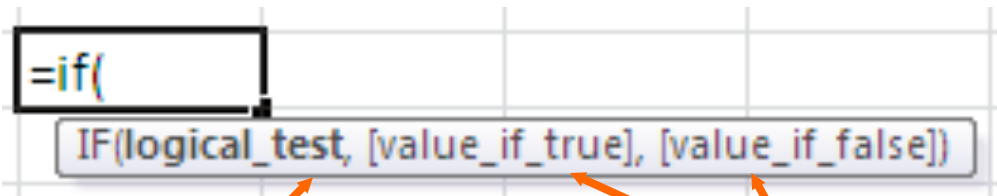
Data type name	Data	Value range	Size
Byte	Small integer	0 to 255	1 byte
Integer	Integer	-32,768 to 32,767	2 byte
Long	Long integer	-2,147,483,648 to 2,147,483,647	4 byte
Single	Short “real” number	$\pm 3.4 \times 10^{38}$ to $\pm 1.4 \times 10^{-45}$	4 byte
Double	Long “real” number	$\pm 1.8 \times 10^{308}$ to $\pm 4.9 \times 10^{-324}$	8 byte

Other Data Types

Data type	Value
Boolean	True, False
String	Text
Date	100/Jan/1 to 9999/Dec/31
Currency	Larger than Long, 922,337,203,477.5808 to 922,337,203,685,477.5807
Other	????

“IF” in Excel Function

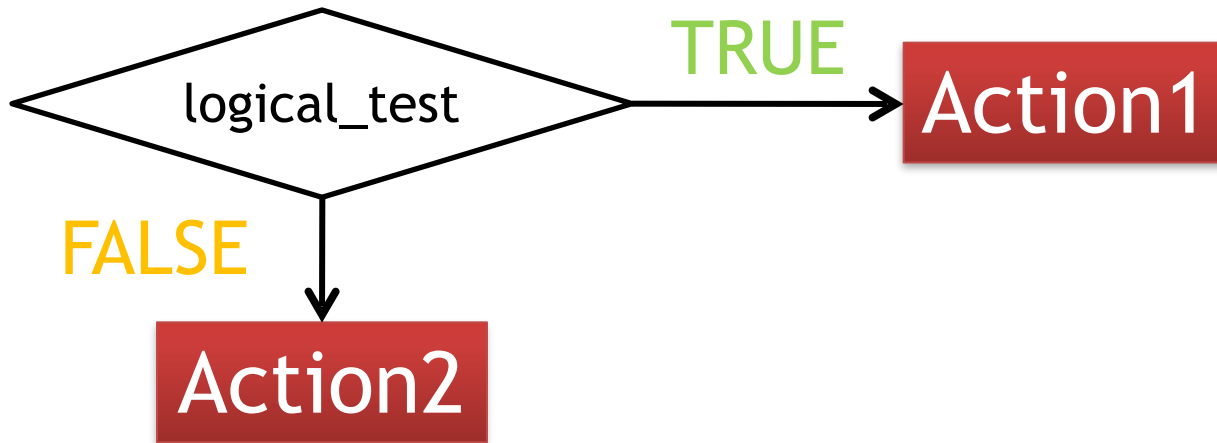
- Branch with “TRUE” or “FALSE”
- IF(logical_test, value_if_true, value_if_false)



Logical formula or Cell number

String with “” or just numbers

If - Then - Else in VBA



```
If logical_test Then
```

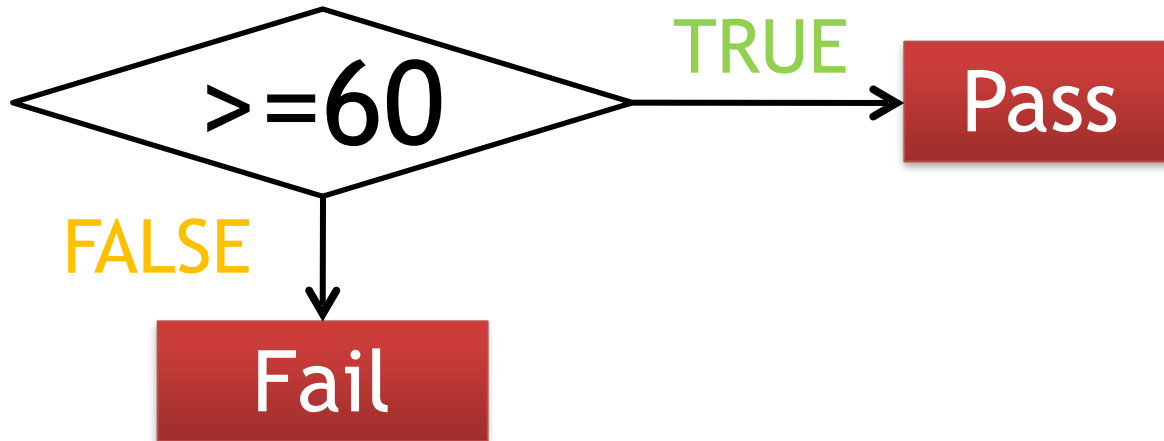
```
    Action1
```

```
Else
```

```
    Action2
```

```
End If
```

Grading in Excel

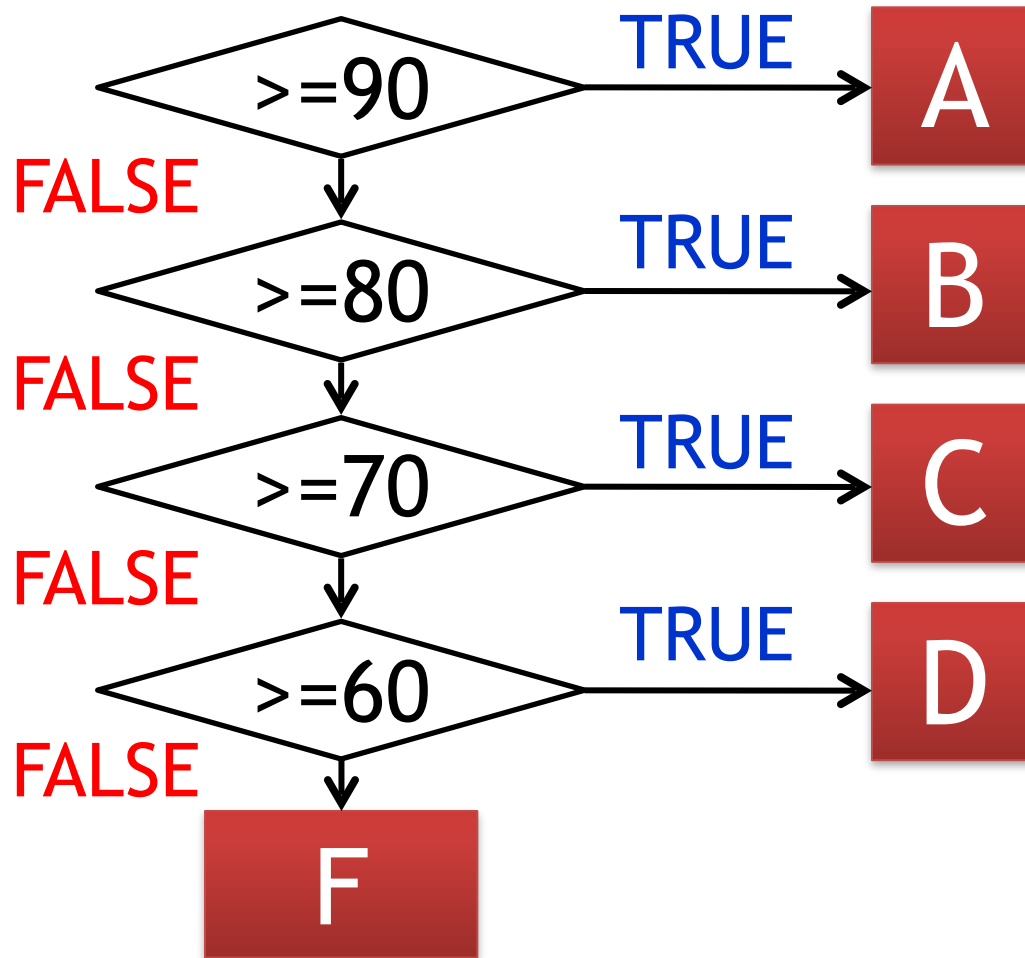


- Grading program in previous lecture
 - Pass if score is more than 60, fail otherwise
- $D16 = \text{IF}(A1 \geq 60, \text{“Pass”}, \text{“Fail”})$

Grading in VBA

```
1 Sub seiseki1()  
2 'Grading Program  
3  
4 Dim score1 As Integer  
5 Dim name1 As String  
6  
7 name1 = InputBox("Enter your name.")  
8 score1 = InputBox("Enter your score.")  
9  
10 If score1 => 60 Then  
11     MsgBox "Congratulations!" & name1 & ", You passed the exam."  
12 Else  
13     MsgBox name1 & ", You failed the exam."  
14 End If  
15  
16 End Sub
```


Nesting “IF” conditions



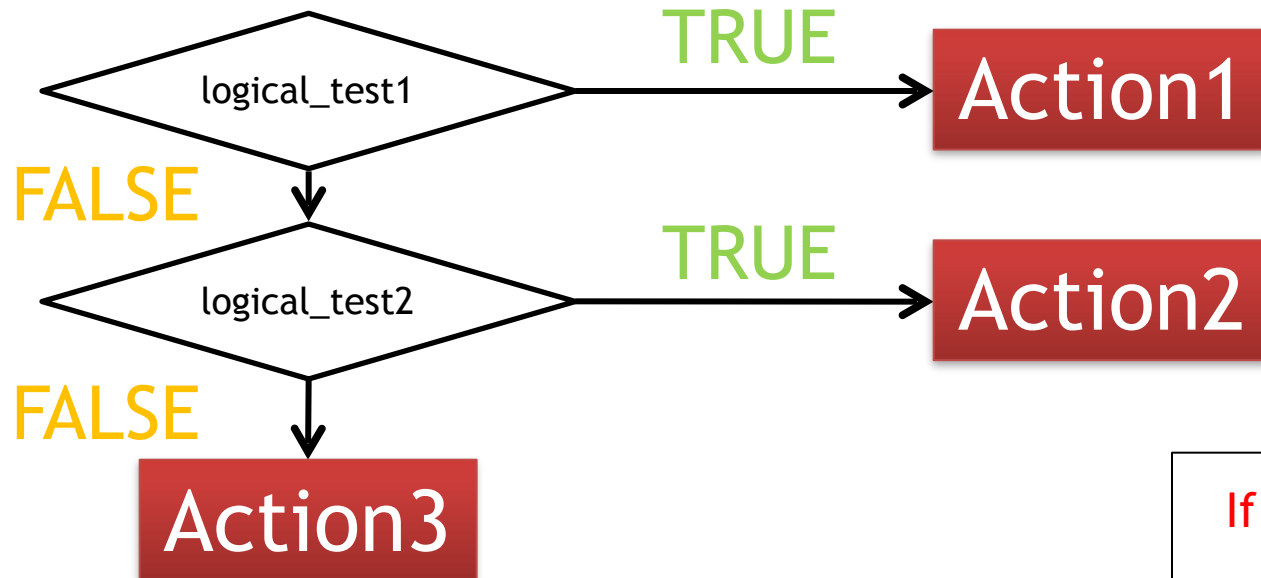
Nesting “IF”s in Excel

- Grade

- A 100 > Score >= 90
- B 90 > Score >= 80
- C 80 > Score >= 70
- D 70 > Score >= 60
- F 60 > Score

```
=IF(B2>=90, "A",  
IF(B2>=80, "B",  
IF(B2>=70, "C",  
IF(B2>=60, "D", "F"))))
```

Nesting “If”s in VBA



```
If logical_test1 Then  
    Action1  
Elseif logical_test2 Then  
    Action2  
Else  
    Action3  
End If
```

Exercise 1

- Transform this excel formula into a VBA program

```
=IF(B2>=90, "A",  
IF(B2>=80, "B",  
IF(B2>=70, "C",  
IF(B2>=60, "D", "F"))))
```

Grading Program If-Then-Else

```
1 Sub seiseki2()  
2 'Grading Program If-Then-Else  
3  
4 Dim score As Integer  
5 Dim name As String  
6  
7 name = InputBox("Enter your name.")  
8 score = InputBox("Enter your score.")  
9  
10 If score >=90 Then  
11     MsgBox name & ", Your grade is A."  
12 ElseIf score >=80 Then  
13     MsgBox name & ", Your grade is B."  
14 ElseIf score >=70 Then  
15     MsgBox name & ", Your grade is C."  
16 ElseIf score >=60 Then  
17     MsgBox name & ", Your grade is D."  
18 Else  
19     MsgBox name & ", Your grade is F."  
20 End If  
21  
22 End Sub
```

情報基礎B (Computer Literacy)

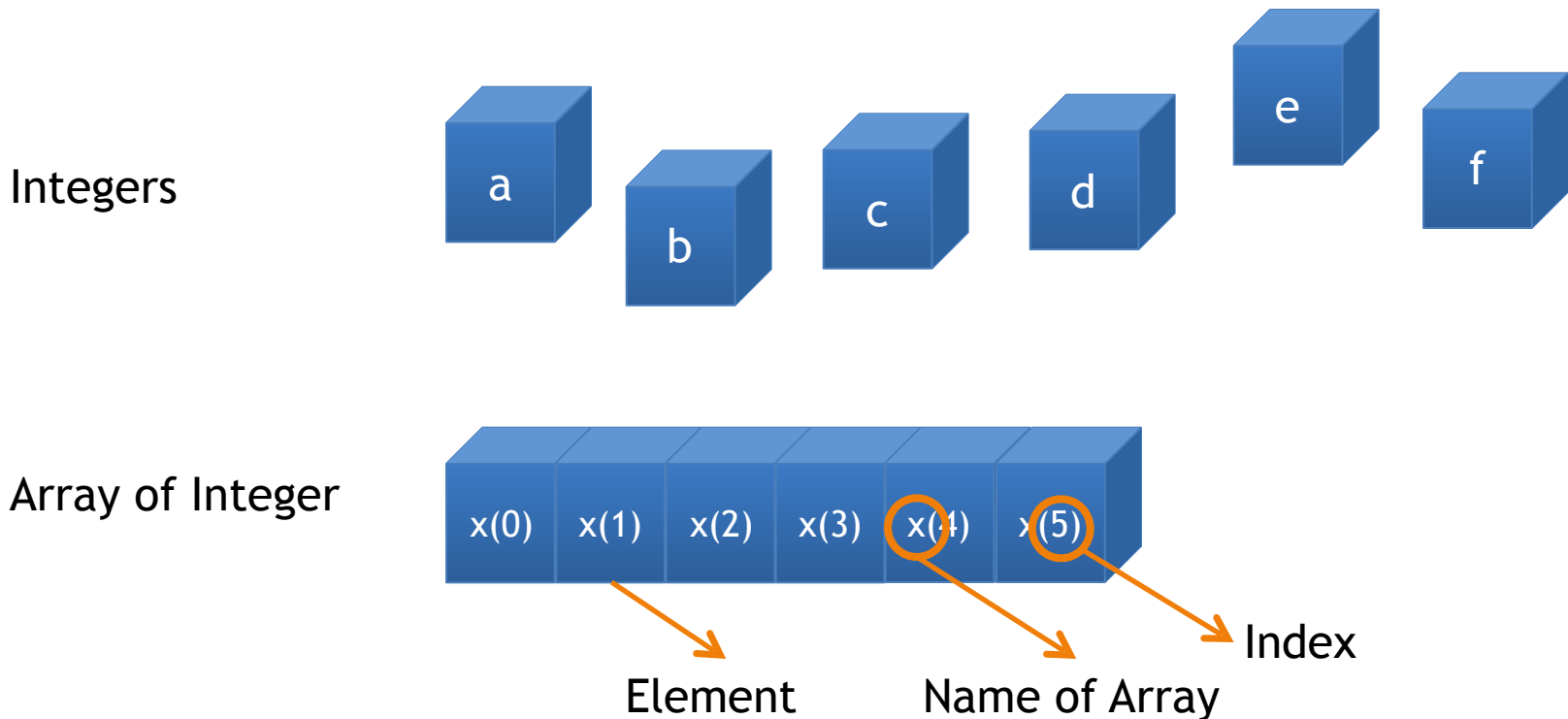
Lecture 8: Arrays and Loops

Matias Korman

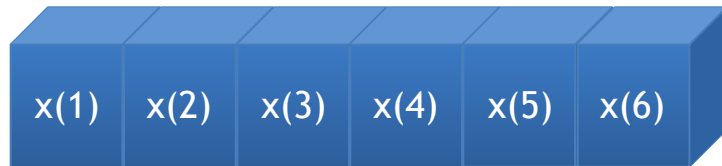
Tohoku University Graduate School of Information Sciences
System Information Sciences
Design and Analysis of Information Systems

Array

- Collection of the same data type
 - Useful for lots of data of same type
 - i.e., grades per student



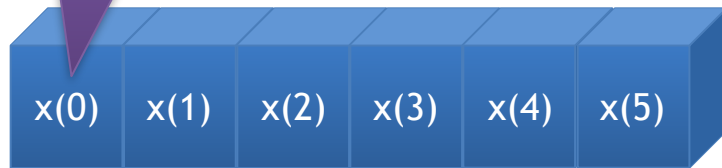
Declaring Arrays



Index starts at "0"



Six boxes to store Integer type variables
Box name: x, Index: 0 to 5

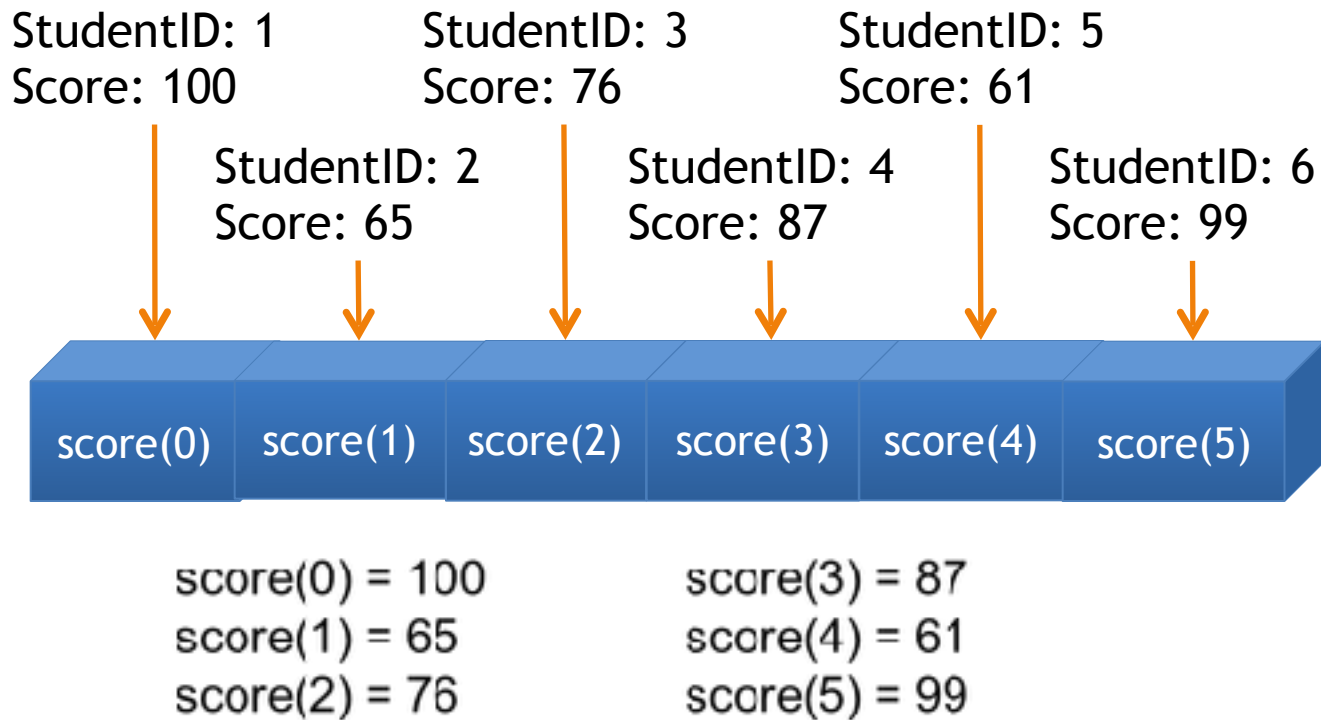


Dim x(5) As Integer

Array name Maximum index Data type

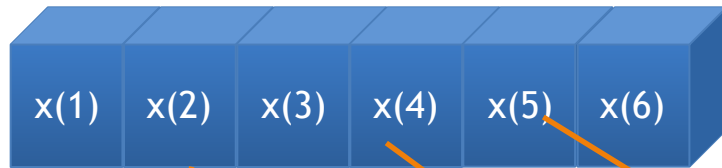
x(5): create six boxes named x each of which contains an Integer

Declaring Arrays



What if I don't have Id 0?

Array Declaration with Index



Element
Box x(1), x(2), ... , x(6)

Array name

Index

Dim x(1 to 6) **As** Integer

Array name

Index Range

Data type

Using arrays in VBA

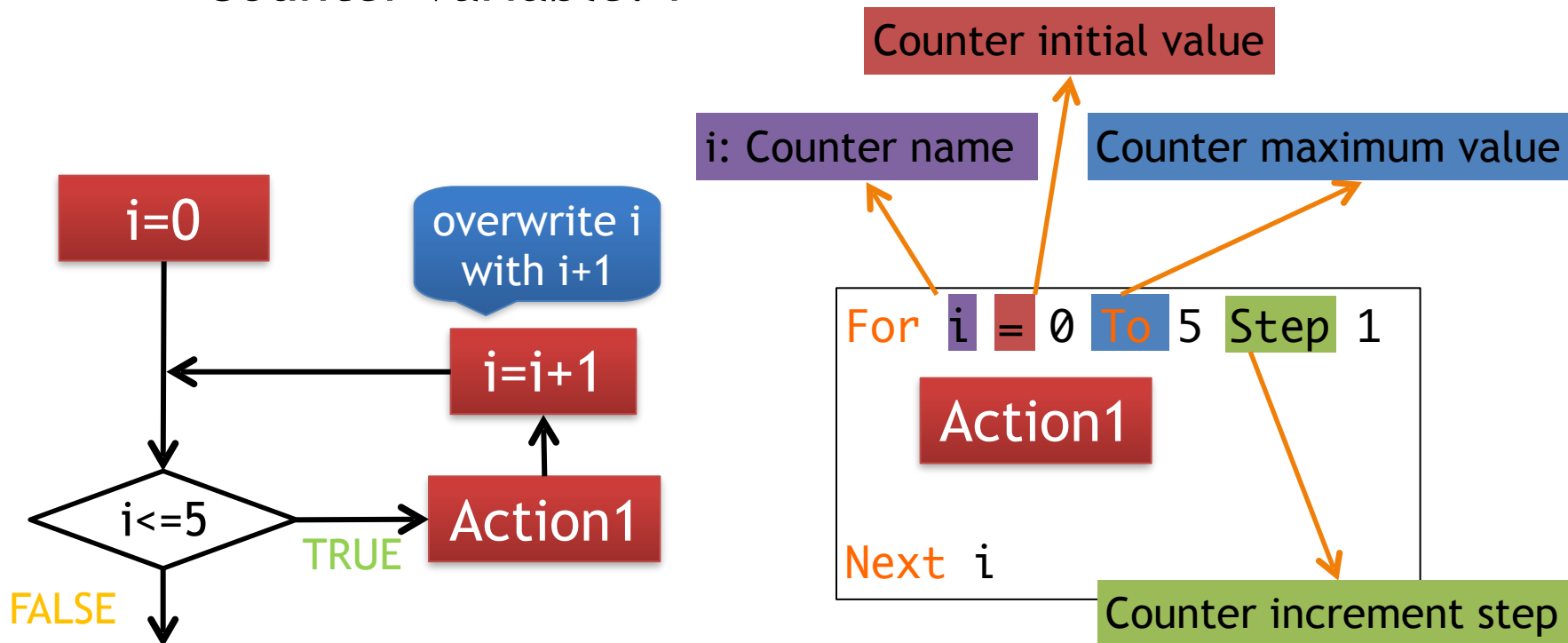
Student ID	Score
1	100
2	65
3	76
4	87
5	61
6	99

```
1 Sub array1()  
2   Dim score(5) As Integer  
3   score(0) = 100  
4   score(1) = 65  
5   score(2) = 76  
6   score(3) = 87  
7   score(4) = 61  
8   score(5) = 99  
9  
10  MsgBox score(0)  
11  MsgBox score(1)  
12  MsgBox score(2)  
13  MsgBox score(3)  
14  MsgBox score(4)  
15  MsgBox score(5)  
16 End Sub
```

Looping



- Repeat same operation several times
 - i.e., compute average grade of each student
 - Use a counter to know when to stop
 - Counter variable: i



What is I?

`Dim i As Integer`

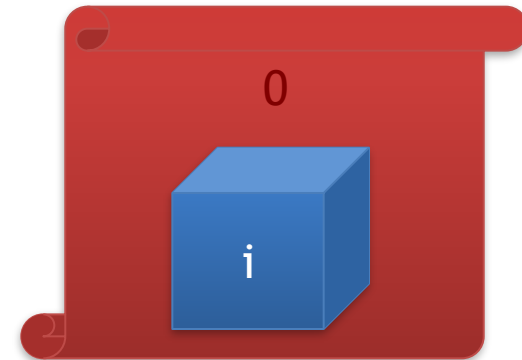
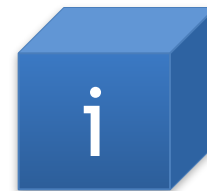


Counter name(arbitrary name): i

- What does $i = i + 1$ mean?
 - Store $i + 1$ onto i
 - i.e., increment i by 1



+ 1 →



```
1 Sub array3()  
2 Display StudentID and score of all students using a loop  
3 'score: array name, i: counter name  
4 Dim score(5) As Integer  
5 Dim i As Integer  
6  
7 score(0) = 100  
8 score(1) = 65  
9 score(2) = 76  
10 score(3) = 87  
11 score(4) = 61  
12 score(5) = 99  
13  
14 For i = 0 To 5 Step 1  
15 MsgBox "StudentID: " & i & ", Score: " & score(i)  
16 Next i  
17 End Sub
```

Student ID	Score
1	100
2	65
3	76
4	87
5	61
6	99

```
14 name(0) = "Koji Tanaka"
15 name(1) = "Hiroshi Abe"
16 name(2) = "Akiko Ito"
17 name(3) = "Ichiro Suzuki"
18 name(4) = "Takako Kato"
19 name(5) = "Junpei Kimura"
20
21 For i = 0 To 5 Step 1
22     MsgBox "StudentID: " & i & ", Name: " & name(i) & ", Score: " &
score(i)
23 Next I
24 End Sub
```

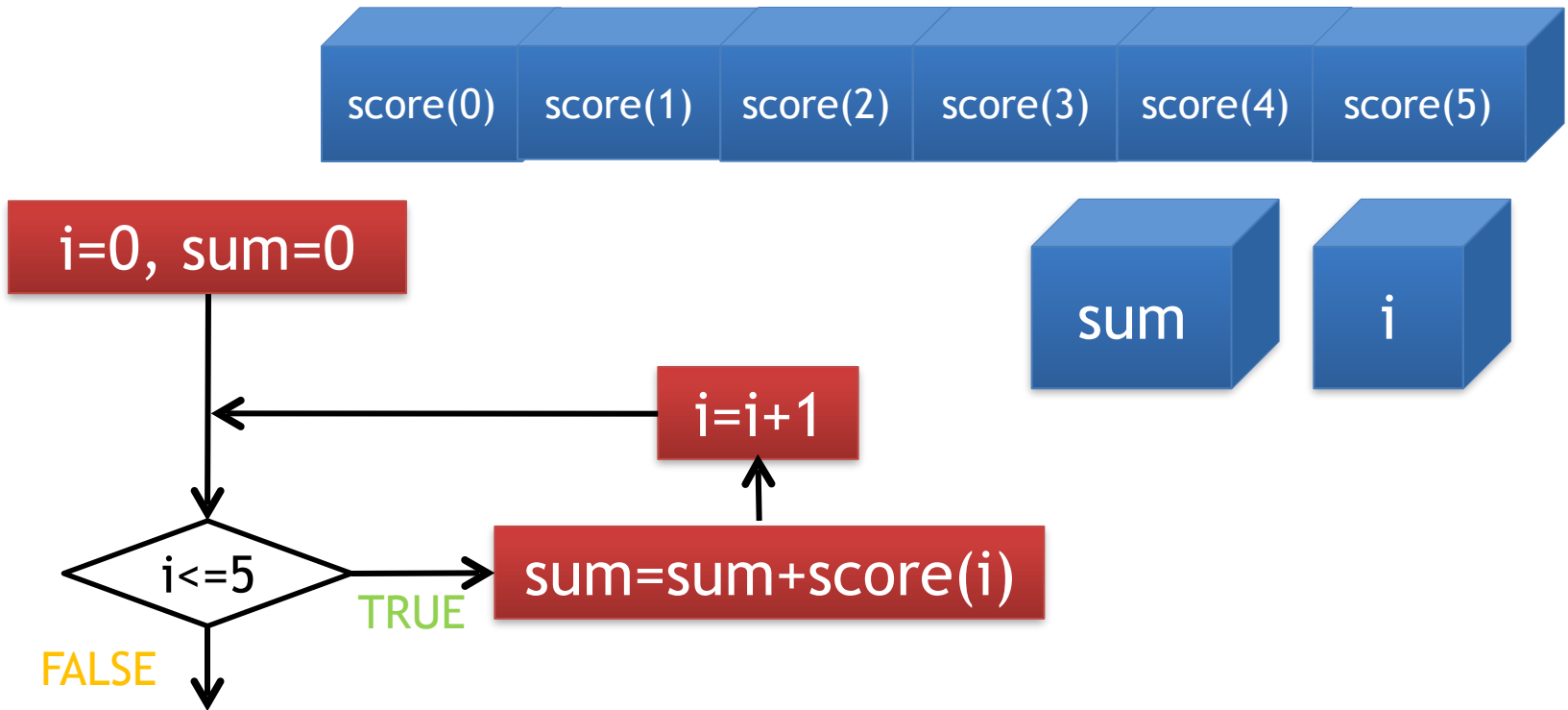
Student ID	name	Score
1	Koji Tanaka	100
2	Hiroshi Abe	65
3	Akiko Ito	76
4	Ichiro Suzuki	87
5	Takako Kato	61
6	Junpei Kimura	99

Exercise 2

- Make similar program that displays for each student if they “Pass” or “Fail” (instead of numerical score)
 - “Pass” only when score is equal or greater than 79
 - “Fail” otherwise
 - Display in increasing order of StudentID
- Ex. StudentID: 1, Name: Koji Tanaka, Score: 100,
 ->Pass

Sum of Scores

- Calculate the sum of score of a student



Exercise 3

- Compute the sum of scores of all students
- Report the sum and the average

```
1 Sub sum()
2 'Calculate the sum of score for all student using For - Next
3 'score: array name, i: counter
4 'sum: variable for sum, ave: variable for average
5 Dim score(5) As Integer
6 Dim i As Integer
7 Dim sum As Integer
8 Dim ave As Single
9 sum = 0
10 ave = 0.0
11 score(0) = 100
12 score(1) = 65
13 score(2) = 76
14 score(3) = 87
15 score(4) = 61
16 score(5) = 99
17 For i = 0 To 5 Step 1
18     sum = sum + score(i)
19 Next i
20 ave = sum/6
21 MsgBox "Sum of score for " & i+1 & "students is " & sum
22 MsgBox "Average is " & ave
23 End Sub
```