"When you come to a fork in the road, take it."
- Y. Berra

Section 5: Branching

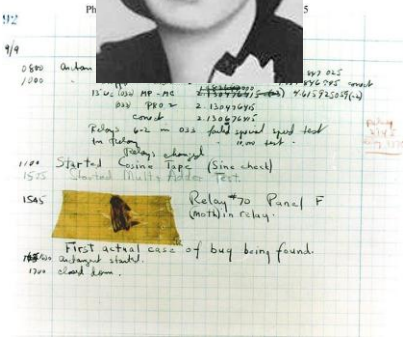
Objectives:

- Structure Charts and Flow Charts
- Branching Instructions and Tracing
- Translating Branches to Java
- Complex Boolean Expressions

138

Historical note …

- 1945: An insect in the circuits blocked the computer Mark I. The computer scientist **Grace Murray Hopper** decided to call any program malfunction « bug »!
- 1951: Invented the first compiler (A0) for generating machine code from a program source code.
- She was one of the main creators of one of the first programming languages: COBOL.



Branching Control Instruction

- So far in the bodies of our algorithms, we have used:
  - a simple statement
  - a straight sequence of simple statements
- Sometimes, we need more than a straight sequence in our solutions, as we sometimes need to do different computations depending on certain conditions.
- Branching instruction (condition)!

140

Problem: Larger of Two Numbers

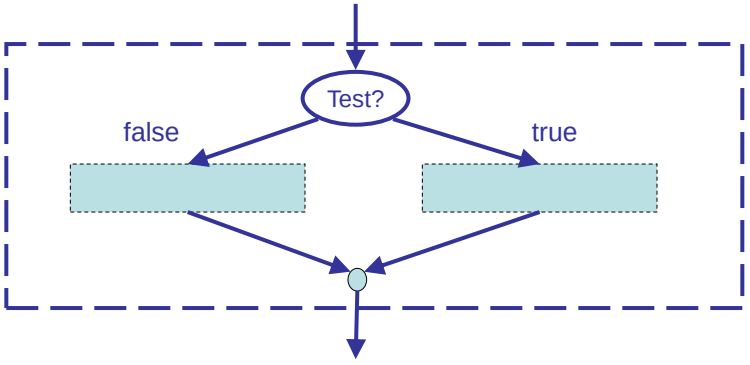
- Write an algorithm to compute the larger of two given numbers.

GIVENS:	x, y	(two numbers)
RESULT:	m	(the larger of x and y)
HEADER:	m $\leftarrow$ max2(x, y)	

- We will represent this with a Branching Control Instruction

141

Branching Control Instruction



- The shaded boxes are INSTRUCTION BLOCKS.
- Note that the branch instruction is complex and can contain many other instructions within its instruction blocks.
  - Graphical representation in our software models!

142

Algorithm Model Diagrams

- Provides visual description of algorithms.
- Composed of nodes connected with arrows.
- Test node:
  - Represents the testing of a condition (Boolean expression with question mark)

- Instruction block :
  - Indicates where another instruction block can be inserted. The instruction block can contain simple or complex instructions (and even no instructions at all)

143

Contents of Instruction Blocks

- Simple instruction (call, assignment)
- Empty statement (Ø = "do nothing")
- Branching control instruction
- Loop control instruction (coming soon…)

- Important: Each block has exactly one entrance (one arrow in) and one exit (one arrow out).

144

Exercise 5-1 - Back to the Larger of Two Numbers

- How would the algorithm for finding the larger (max) of two values x and y be written in a model diagram ?

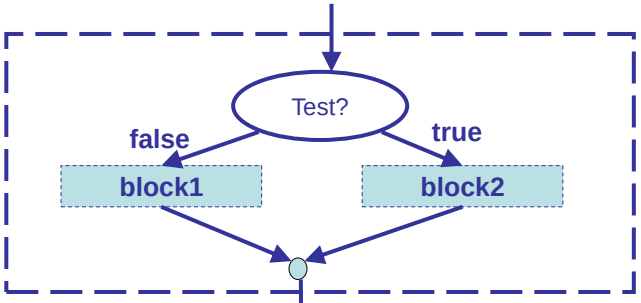
145

Exercise 5-2 - Maximum of 3 numbers ?

- Given three numbers x, y, and z, find the maximum of the three values.
  - Version 1: nested tests
  - Version 2: sequence of tests

146

Translating Branches to Java



- Java :

```
if (test)
{
   // Instructions
}
else
{
   // Instructions
}
```

Instruction block 2

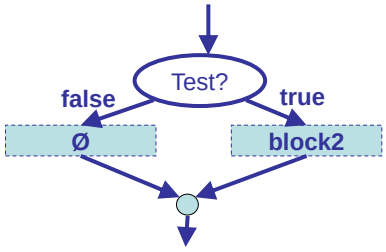
Instruction block 1

147

Java if Instruction - Option

The else part of the instruction is optional. Thus in algorithms use only the empty statement block in the false part of the if statement.

```
if (test)
{
    // Instructions
}
else
{
    // Instructions
}
```



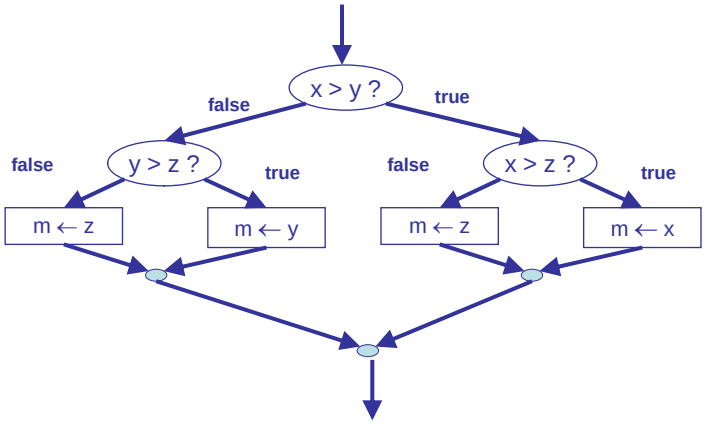

```
if (test)
{
    // Instructions
}
```


Exercise 5-3 – Translating Branches

Givens: x, y, z (three numbers)
Result: m (the largest given value)
Header: m←max3(x, y, z)

- Two solutions:
  - sequence of branch instructions
  - nested branch instructions
- Translate the latter into Java:

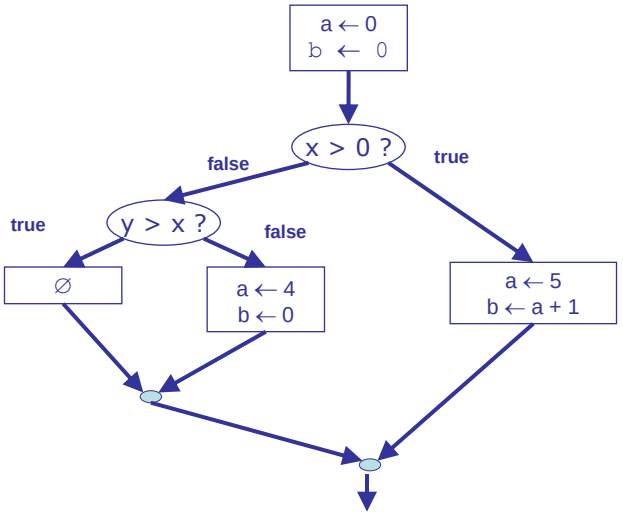
Nested Branches



Exercise 5-4 - Translation of Nested Branches

?

Example of an instruction block with a branch instruction



152

Exercise 5-5 – Translation of an Instruction Block

?

153

Tracing algorithms with branching

- When tracing an algorithm with tests (branches or loops)
  - Number the tests as well as the statements
  - Include a row in the trace indicating which test is being done and whether it is true or false.
  - Indicate only the instructions executed in the trace.

Exercise 5-6 - Trace of Maximum of 3 numbers

Trace: max3(5, 11, 8)

Exercise 5-7 - Movie Tickets

- Calculate the amount to charge for a person's movie ticket given that the charge is $7 for a person 16 or under, $5 for a person 65 or older, and $10 for anyone else.
  - Version 1: nested tests
  - Version 2: sequence of tests

Boolean Variables

- A Boolean variable is one which can have only 2 possible values: TRUE or FALSE.
  - In reality represented by two values (e.g. 0 and 1), but in high level language use only these key words are allowed!
- An assignment statement is used to put a value into a Boolean variable, e.g.,

  x ← TRUE

  y ← FALSE
- The outcome of a test (Boolean expression) can be assigned to a Boolean variable:

  x ← (a < 0)

Exercise 5-8 - Positive Value

- Write an algorithm which checks if a given number x is positive.

Compound Boolean Expressions

- A compound Boolean expression consists of two or more Boolean expressions connected by operators AND and/or OR.
- Exercise 5-9 - Write a compound Boolean expression that is true if a given age is between 16 and 65 (not including 16 or 65) and false otherwise.

Truth Tables

- A TRUTH TABLE for a compound Boolean expression shows the results for all possible combinations of the simple expressions:

x	y	x AND y	x OR y
TRUE	TRUE		
TRUE	FALSE		
FALSE	TRUE		
FALSE	FALSE		

160

Operator NOT

x	NOT x
TRUE	FALSE
FALSE	TRUE

- NOT is an operator to negate the value of a simple or compound Boolean expression:
- Example. Suppose age = 15. Then:
  - Expression age > 16 has a value FALSE, and NOT (age > 16) has a value TRUE.
  - Expression age < 65 has a value TRUE, and NOT (age < 65) has a value FALSE.

161

Exercise 5-10
More Compound Boolean Expressions

Suppose x = 5 and y = 10.

Expression	Value
(x > 0) AND (NOT (y = 0))	
(x > 0) AND ((x < y) OR (y = 0))	
(NOT (x > 0)) OR ((x < y) AND (y = 0))	
NOT ((x > 0) OR ((x < y) AND (y = 0)))	

Expressions in Tests

- The TEST in a Branch or Loop may be any Boolean expression:
  - Boolean variable
  - Negation of a Boolean expression
    - NOT (Java: `!`)
  - Comparison between two values
    - Java operators: `==` `!=` `<` `>` `<=` `>=`
    - The data being compared may not necessarily be `boolean`, but the result of the comparison is `boolean`
  - Join two Boolean expressions
    - AND (Java: `&&`)
    - OR (Java: `||`)
- Watch out for
  - confusing `=` with `==`
  - confusing AND with OR
- e.g. test if x is in the range 12..20:

```
(x >= 12) && (x <= 20)
```