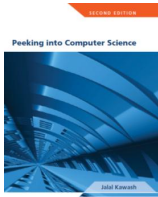


3 Programming



Peeking into Computer Science

© Jalal Kawash 2010

1

- Mandatory: Chapter 5 – Section 5.5
- Jalal's resources:
 - "How to" movies and example programs available at:
<http://pages.cpsc.ucalgary.ca/~kawash/peeking/alice-how-to.html>
- JT's resources:
 - www.cpsc.ucalgary.ca/~tamj/203/extras/alice



Reading Assignment

Peeking into Computer Science

© Jalal Kawash 2010

2

- To simulate the possibility of different events occurring as a program is a run a random number function can be employed.
- Typically the function generates a number within a certain range:
 - 1 – 2 (coin flip)
 - 1 – 6 (normal die roll)
 - 1 – 100 (percentage)
 - Etc.

JT's Extra: Random Numbers

Peeking into Computer Science

© Jalal Kawash 2010

Example 9: random numbers

- It can be invoked by the random function of class 'World'

The only values generate: 1, 2, 3, 4, 5, 6
(no real numbers)

```
result - set value to random number minimum = 1 maximum = 7 integerOnly = true more...
DiceMan - say result as a string more...
```

Inclusive


Exclusive

The screenshot shows a 'world' object with a 'random' function highlighted in red. The function is defined as 'choose true probabilityOfTrue of random number'. The 'random' function is also listed in the 'functions' panel on the right side of the interface.

JT's Extra: Dice Example

Peeking into Computer Science

© Jalal Kawash 2010



If-Else Statements
Alice makes decisions

5

At the end of this section, you will be able to:

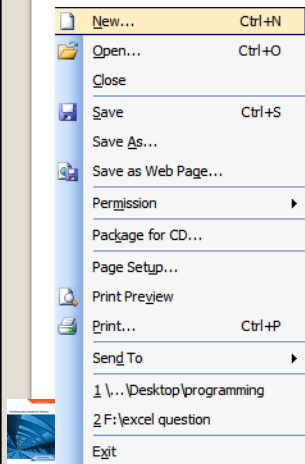
1. Understand conditional execution
2. Understand and use the Alice if/else statement
3. Use functions as arguments to other methods
4. Use conditional structures in flowcharts

Objectives

Peeking into Computer Science

© Jalal Kawash 2010

- Frequently this is needed when the program must react to user input or the actions of the user.



New: series of instructions

Open: series of instructions

Save: series of instructions

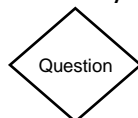
Etc.

JT's Extra: Choosing Among Alternative Course Of Action

Peeking into Computer Science

© Jalal Kawash 2010

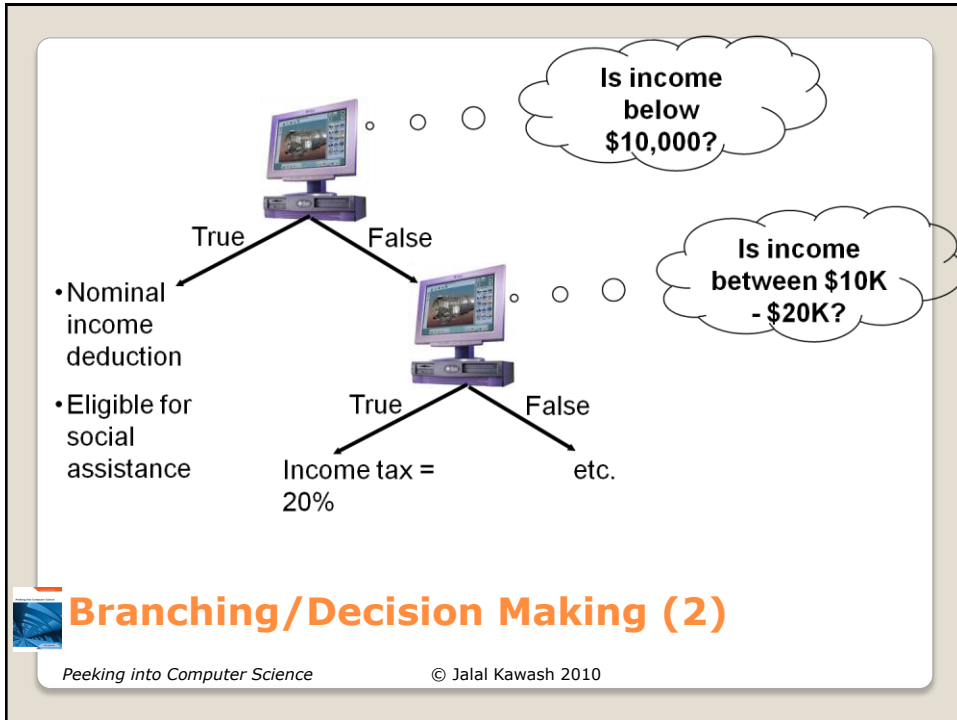
- It's the mechanism used in many programming languages used to execute different instructions when alternatives are faced.
- At the branch (where the decision must be made) a logical statement determines which alternative occurs.
 - Another way of looking at it: a question is asked which has a true or false answer.
 - The answer to the question determines which branch executes.
- The flowchart symbol used to represent a branch/decision is a diamond.



Peeking into Computer Science

© Jalal Kawash 2010

Branching/Decision Making



- Programs need to make decisions
- An ATM program must verify your password to know how to proceed
- This is called *conditional Execution*
- The execution of a group of statements is conditional on a Boolean condition

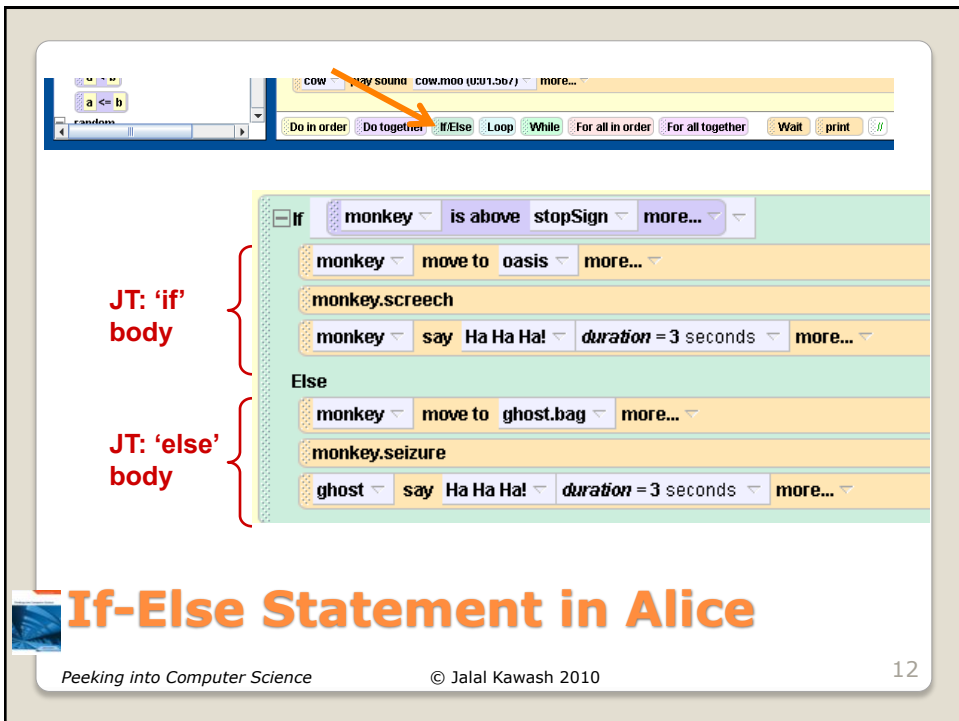
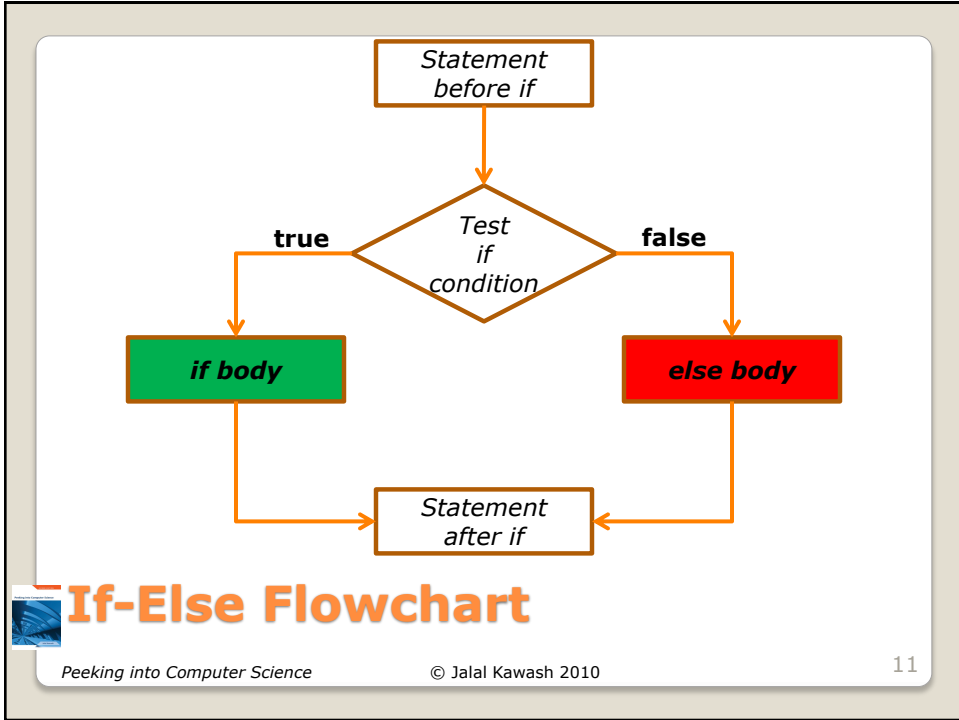
If the password is correct then
go to the next step

Else
give an error message

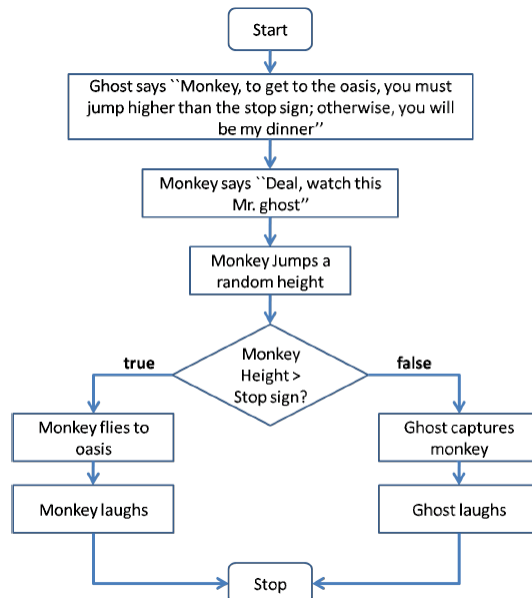
JT (new terminology): body is the part that executes only under a certain condition

Conditional Execution

Peeking into Computer Science © Jalal Kawash 2010



Example Story Line



Peeking into Computer Science

© Jalal Kawash 2010

13

ghost say Monkey, To get to the oasis, you must jump higher than the stop sign

monkey say Deal, watch this Mr. ghost! duration = 3 seconds more...

monkey.monkey.Jump

monkey move up random number minimum = 0.1 maximum = 4 n

random number minimum = 0.1 maximum = 4 more...

Returns a random number between 0.1 and 4
value substituted for distance parameter of move

If monkey is above stopSign more...

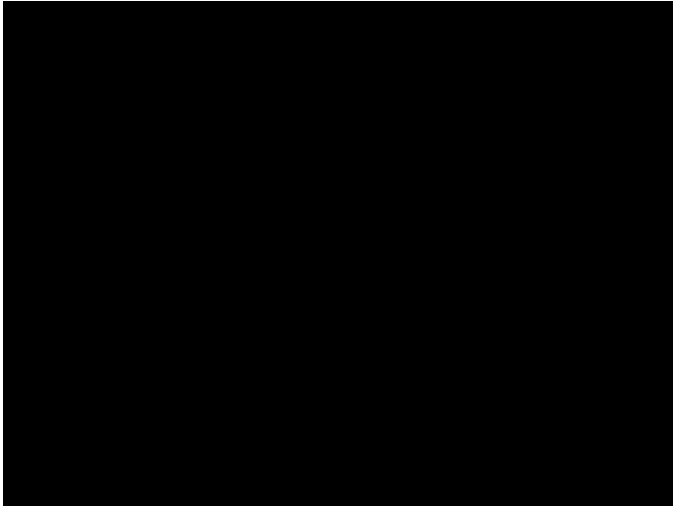
Returns a Boolean value

Use of Functions

Peeking into Computer Science

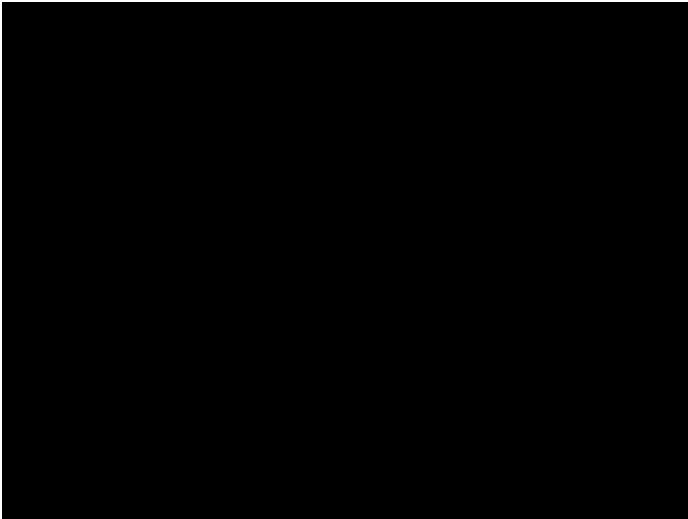
© Jalal Kawash 2010

14



When the if condition is true

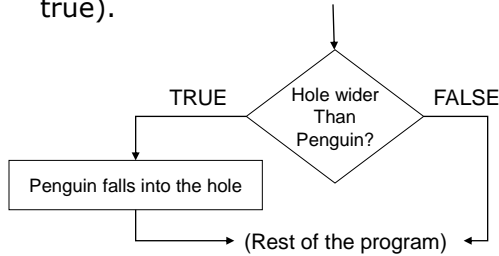
Peeking into Computer Science © Jalal Kawash 2010 15



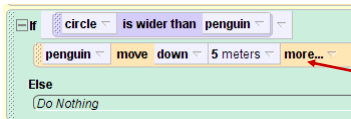
When the if condition is false

Peeking into Computer Science © Jalal Kawash 2010 16

- If you want the program to react when the logical statement evaluates to true (the answer to the question is true).



- Just use the 'if' and leave the 'else' portion empty.



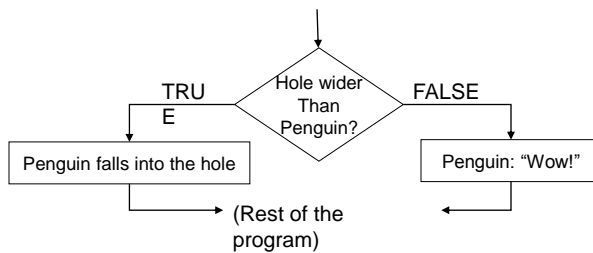
Indented part after the 'if': what executes if the statement is true

JT: One Action: Do Something When True

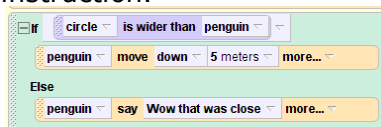
Peeking into Computer Science

© Jalal Kawash 2010

- You want the program to do something when the logical statement evaluates to true and something different when the statement evaluates to false.



- Add instructions for both the 'if' and the 'else' portion of the instruction.

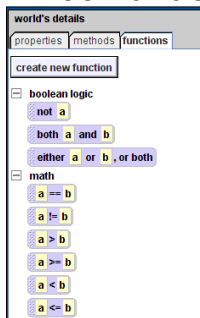


JT: Do Something For Both True/False Cases

Peeking into Computer Science

© Jalal Kawash 2010

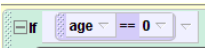

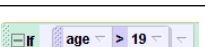
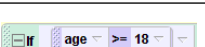

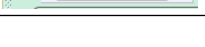
- Are used in conjunction with logical statements and relate or compare things.
- The comparison will result in a true or false result.
 - E.g., temperature is above freezing (greater than zero).
- Accessing the logical operators in Alice.
 - Look under the functions of the 'world' object.



JT: Logical Operators

Peeking into Computer Science

© Jalal Kawash 2010

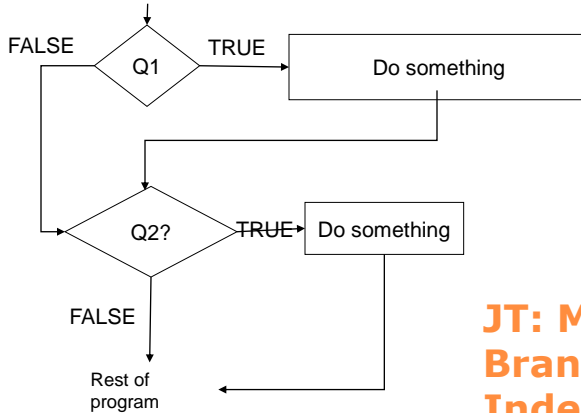
Desired comparison	Operator (Alice)	Example
Equality	==	
Not equal to (inequality)	!=	
Greater than	>	
Greater than, equal to	>=	
Less than	<	
Less than, equal to	<=	

JT: Types Of Logical Operators

Peeking into Computer Science

© Jalal Kawash 2010

- If a number of questions must be asked by a program and the answer to one question has no impact on the answer to the other questions.
 - That means that all questions will be asked regardless of whether any questions answered true or false.

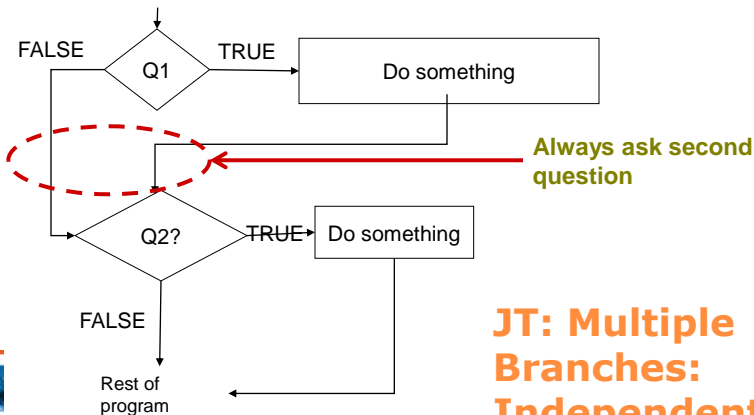


**JT: Multiple
Branches:
Independent**

Peeking into Computer Science

© Jalal Kawash 2010

- If a number of questions must be asked by a program and the answer to one question has no impact on the answer to the other questions.
 - That means that all questions will be asked regardless of whether any questions answered true or false.



**JT: Multiple
Branches:
Independent**

Peeking into Computer Science

© Jalal Kawash 2010

Example 10: multiple successive branches

First Boolean question: Always asked

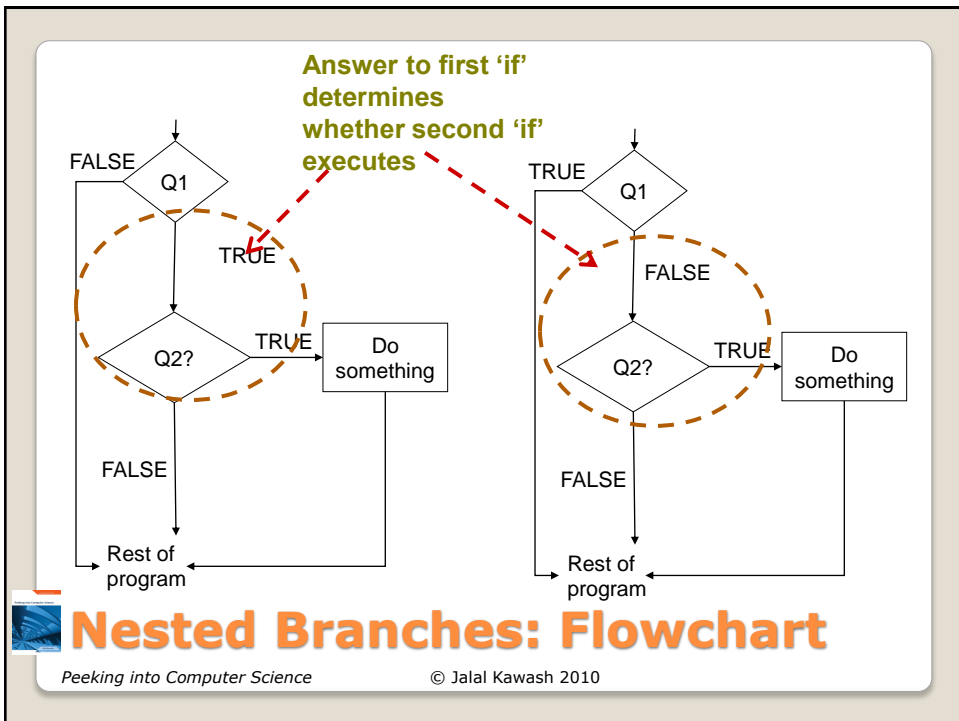
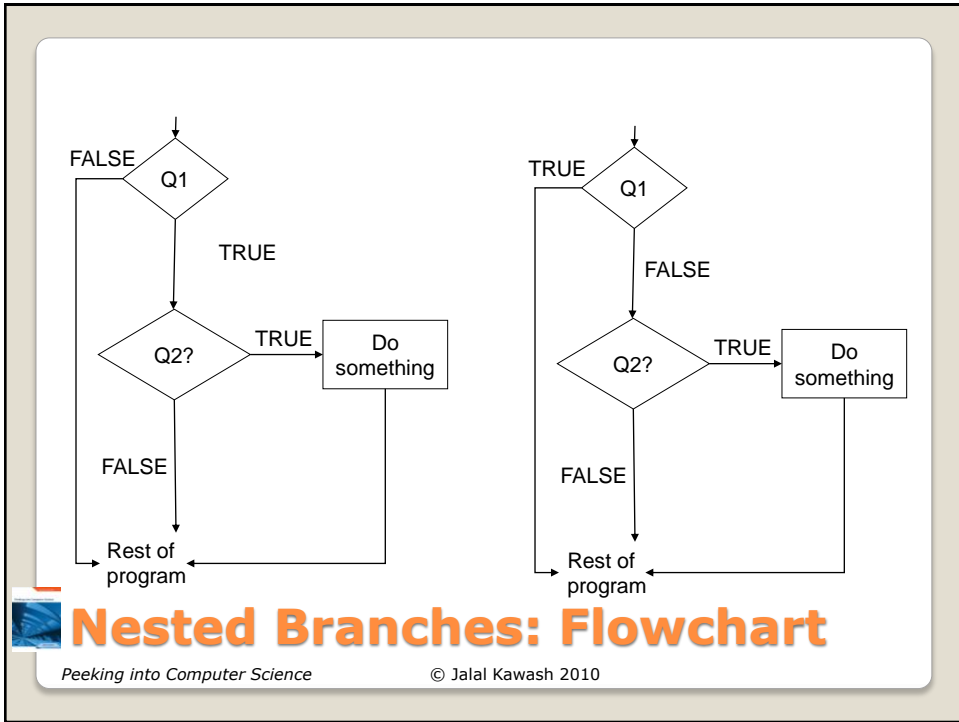
Second Boolean question: Not indented, always asked

JT's Extra: Multiple Branches

Peeking into Computer Science © Jalal Kawash 2010 23

- Sometimes the answer to one question will determine whether or not other questions will be asked.
 - The answer to one question (true or false) has an effect on whether another question is asked.
 - The asking of the second question is dependent on the answer to the first question.
 - (Alternatively: the second question is nested inside the first question).

Nested Branches



- Multiple if/else statements can be used in succession.
- Example program 11: nested_branches

Indented: executes if first 'if' is true

Also indented: execution is dependent on the first 'if'

(Only ask about blue if user does not indicate that red is the favorite)

Indented twice: only executes if other cases answer false

JT: Multiple Branches: Dependent (RHS Case)

Peeking into Computer Science


© Jalal Kawash 2010

JT: Multiple Branches: Dependent (LHS Case: Nonsensical)

Peeking into Computer Science

© Jalal Kawash 2010

28



Loops
Do it again

29

At the end of this section, you will be able to:

1. Understand repetition structures
2. Use the loop statement
3. Use the while statement
4. Understand when to use a loop or a while
5. Use loop structures in flowcharts

Objectives

Peeking into Computer Science

© Jalal Kawash 2010

- A *loop* statement allows a group of statements to be repeated for a number of times

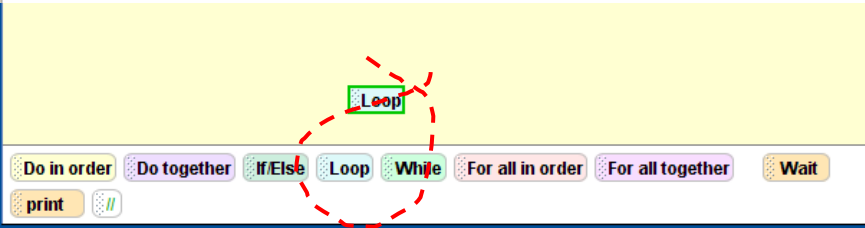
loop for 5 times

```

monkey.monkeyJump
monkey.move(up,1 meter)
monkey.move(down,1 meter)

```

Loop Statement



The screenshot shows a palette of programming blocks. A red dashed circle highlights the 'Loop' and 'While' blocks. The blocks in the palette are: 'Do in order', 'Do together', 'If/Else', 'Loop', 'While', 'For all in order', 'For all together', 'Wait', 'print', and '//'. The 'Loop' and 'While' blocks are highlighted with a red dashed circle.

JT's Extra: Adding A Loop In Alice

The screenshot shows the Alice programming environment. At the top, there is a control panel with a dropdown menu set to "Loop 5 times" and a "times" field. To the right is a button labeled "show complicated version". Below this, the code area contains the following sequence of actions:

- `monkey.monkeyJump`
- `monkey` dropdown, `move up` dropdown, `1 meter` dropdown, `more...` dropdown
- `monkey` dropdown, `move down` dropdown, `1 meter` dropdown, `more...` dropdown

Loop Statement in Alice

Peeking into Computer Science © Jalal Kawash 2010 33

```

    graph TD
      Start([Start]) --> Loop
      subgraph Loop [Loop 5 times]
        direction TB
        Jumps[Monkey Jumps] --> Up[Monkey moves up 1m]
        Up --> Down[Monkey moves down 1m]
      end
      Loop --> Stop([Stop])
  
```

The flowchart starts with a "Start" terminal, followed by a loop structure labeled "Loop 5 times". Inside the loop, the process flow is: "Monkey Jumps", then "Monkey moves up 1m", and finally "Monkey moves down 1m". The loop ends at a "Stop" terminal.

Loop in Flowcharts

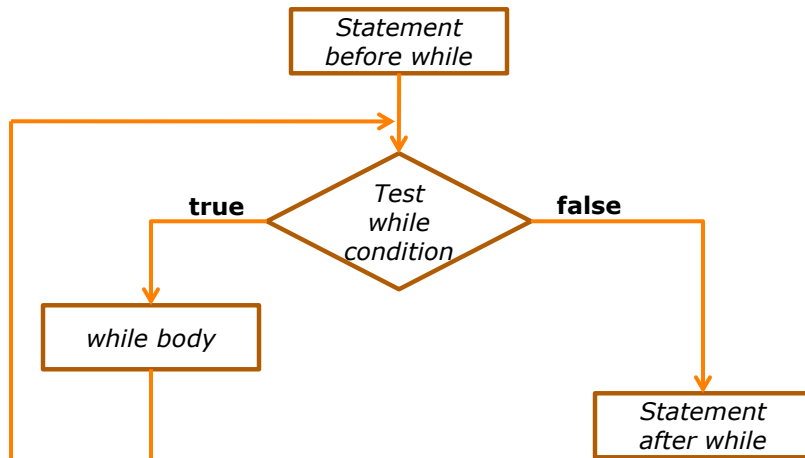
Peeking into Computer Science © Jalal Kawash 2010 34

- Loops are used to repeat a part of the program (or potentially the entire program itself).
- They are similar to branches where a logical statement is employed.
- In the case of a loop if the logical statement evaluates to true then a part of the program will repeat (loop).
- The repetition will continue until the logical statement evaluates to false.

JT's Extra: Boolean-Controlled Loops

Peeking into Computer Science

© Jalal Kawash 2010



While Statement

Peeking into Computer Science

© Jalal Kawash 2010

36

Example 11

The image shows a Scratch script with the following blocks:

- count** set value to **1**
- troll** say **Me play bam!**
- While** loop with condition **count <= 5** containing:
 - troll.head.skullCap** say **count** as a string
 - troll.rightArm** roll left 0.25 revolutions
 - Do together** block:
 - troll.rightArm** roll right 0.5 revolutions
 - ground** think **Ouch!** with **fontSize = (count * 8)**
 - troll.rightArm** roll left 0.25 revolutions
 - increment count** by 1

Annotations:

- Starting value**: points to the '1' in the 'set value to' block.
- Condition for repeating**: points to the '<= 5' in the 'While' loop condition.
- Loop 'body': Instructions that are repeated**: a red bracket encompasses the entire 'While' loop block.

JT's Extra: Counting With While

Peeking into Computer Science © Jalal Kawash 2010 37

```

    graph TD
      START([START]) --> Init[count = 1]
      Init --> Decision{count <= 5?}
      Decision -- True --> Body[Troll counts, arm moves up]
      Body --> Parallel[Arm moves down | Ground cries out]
      Parallel --> Increment[Arm moves up, count = count + 1]
      Increment --> Decision
      Decision -- False --> END([END])
    
```

The flowchart starts at **START**, followed by the process box **count = 1**. It then reaches a decision diamond **count <= 5?**. If **True**, it enters a loop: **Troll counts, arm moves up**, followed by a parallel box for **Arm moves down** and **Ground cries out**, then **Arm moves up, count = count + 1**, which loops back to the decision diamond. If **False**, it proceeds to **END**.

JT's Extra: Example Flowchart

Peeking into Computer Science © Jalal Kawash 2010 38

While Example


Peeking into Computer Science © Jalal Kawash 2010 39

Example 12

If-Else branch is part of the body of the loop

JT's Extra: Combining Loops And Branches

Peeking into Computer Science © Jalal Kawash 2010 40



While Example

Peeking into Computer Science

© Jalal Kawash 2010

41