ITI 1120
Lab #4

Loops

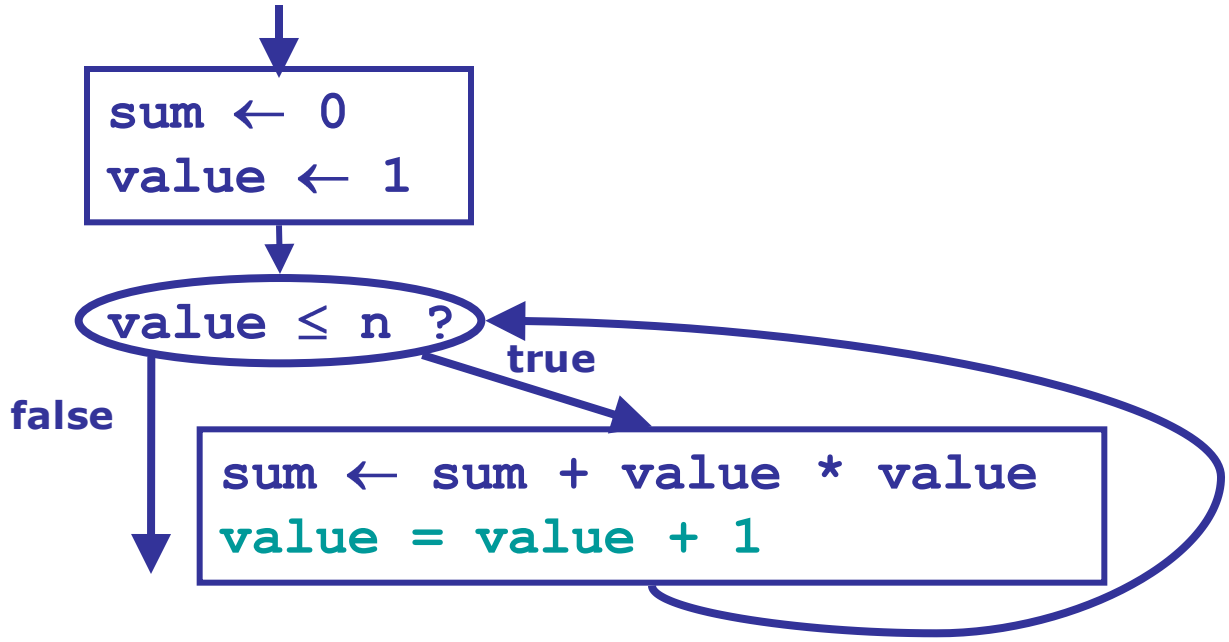
Contributors: G. Arbez, M, Sehatkar, M. Eid, D. Inkpen, A. Williams, D. Amyot

For today's lab:

- While statement
- For loop
- Do/While Loop
- Exercises

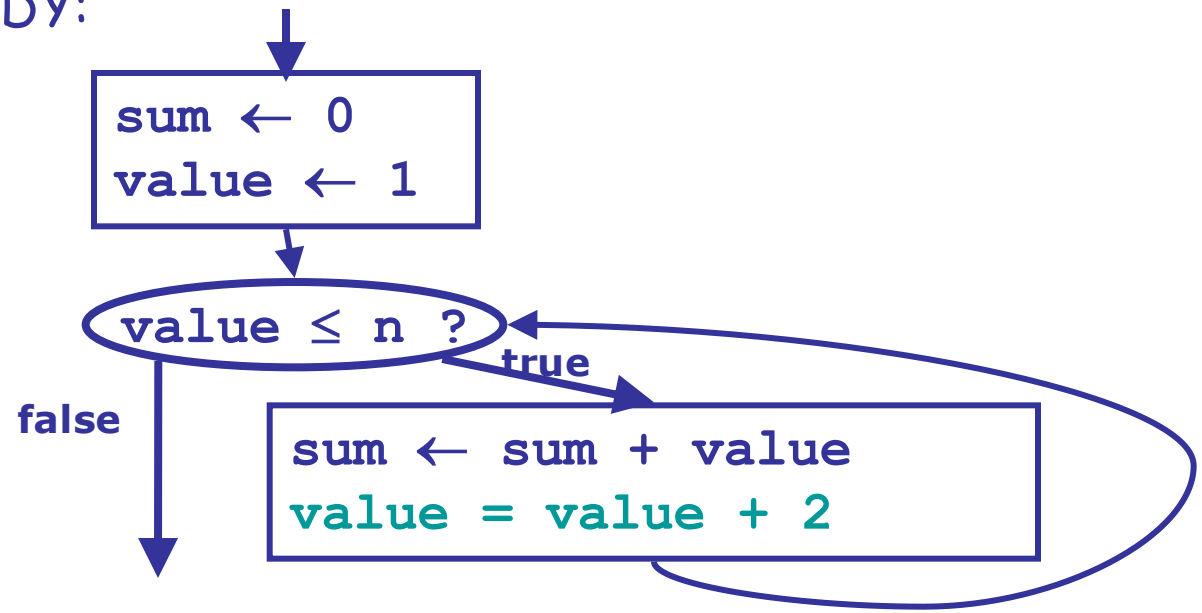
Sum of Squares

GIVENS:	n	(a number ≥ 1)
RESULTS:	sum	(sum of squares from 1^2 to n^2)
INTERMEDIATES:	value	(current value to square)
HEADER:	sum ← sumOfSquares(n)	
BODY:		



Sum of Odd Integers

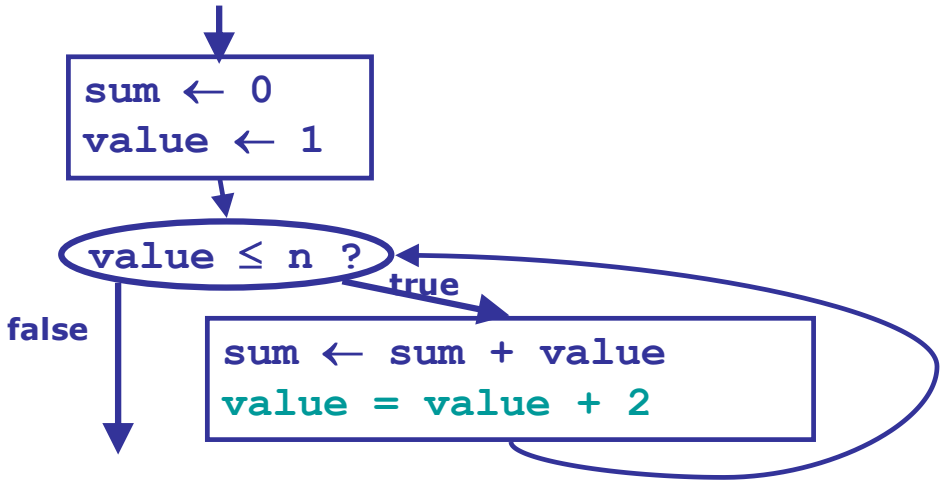
GIVENS:	n	(a number ≥ 1)
RESULTS:	sum	(sum of odd integers up to n)
INTERMEDIATES:	value	(current odd integer)
HEADER:	sum ← sumOddIntegers(n)	
BODY:		



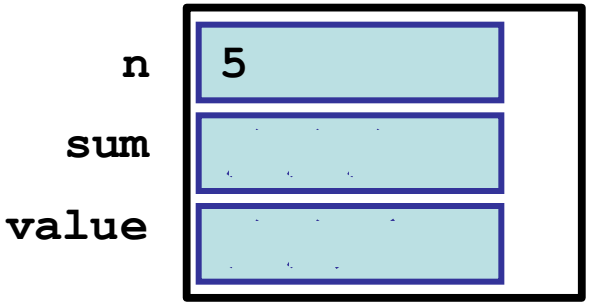
Programming Model for Tracing sumOddIntegers(5)

Program Memory

GIVENS: n (a number ≥ 1)
RESULTS: sum (sum of odd integers up to n)
INTERMEDIATES: value (current odd integer)
HEADER: sum ← sumOddIntegers(n)
BODY:



Working Memory



CPU

Trace for Sum of Odd Integers

Statement	n	value	sum
Initial values	5	?	?
sum ← 0			0
value ← 1		1	
index <=n ? (1 <= 5) true			
sum ← sum + value			1
Value = Value + 2		3	
value <=n ? (3 <= 5) true			
sum ← sum + value			4
Value = Value + 2		5	
value <=n ? (5 <= 5) true			
sum ← sum + value			9
Value = Value + 2		7	
value <=n ? (5 <= 5) false			

Exercise 1

- The code below is supposed to print the integers from 10 to 1 backwards. You will find the code in Lab4Ex1.java.
  - You need to find 2 logical errors in the code.
  - Take the time to draw an algorithm body to follow the logic and find the errors BEFORE any coding. Use the programming model to trace.
  - Correct the code, compile and run to check out your answer.

```
count = 10;
while (count >= 0)
{
 System.out.println(count);
 count = count + 1;
}
```

Exercise 2

- Develop a program to print all numbers from 1 to n inclusively.
  - First create algorithms using Word (use Visio to create the algorithm body that uses loops). Start with the template Lab4Ex2.doc.
  - Then translate your algorithms into a Java program and test. Start with Lab4Ex2.java.
  - The main algorithm/method obtains from the user the value for n and calls the problem solving algorithm.
  - The problem solving algorithm/method prints to the screen the values from 1 to n.

Exercise 3

- The Fibonacci series are the numbers in the following sequence:

  0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, …

- By definition, the first two numbers in the Fibonacci series are 0 and 1, and each successive number is the sum of the previous two, e.g.

  1 = 0 + 1, 2 = 1 + 1, 3 = 1 + 2, 5 = 2 + 3, etc.

- Write a program that asks the user for a positive integer *n* and computes and prints all the numbers in Fibonacci series up to n. If *n* is a number in Fibonacci series then the last number in the sequence which the program will print is *n*, otherwise it will print the largest number in the series which is less than *n*.

Exercise 3 (Cont.)

- Develop your algorithms with Visio and Word – start with Lab4Ex3.doc.
- Translate the algorithms to Java. Start with Lab4Ex3.java and complete the main and problem solving methods.
- The main algorithm/method prompts the user to enter a positive integer and then calls fibonacci() method to compute and print the Fibonacci series. Then it asks the user for another run. If the user decides to compute Fibonacci series for another *n*, then the program repeats the steps described, otherwise the program terminates.

Exercise 4-a

- The *factorial* of n (written n!) is the product of the integers between 1 and n. Thus 4! = 1*2*3*4 = 24. By definition, 0! = 1. Factorial is not defined for negative numbers.
- Write a program that asks the user for a non-negative integer and computes and prints the factorial of that integer. Computing the factorial is a lot like computing the sum of integers from 1 to n, but uses multiplication instead of addition. Don't forget about dealing with 0.
  - Develop your algorithms with Visio and Word – start with Lab4Ex4.doc.
  - Translate the algorithms to Java. Start with Lab4Ex4a.java and complete the problem solving method.
  - The main algorithm/method prompts the user to enter a positive integer, calls calcFact() to get its factorial, and prints the results the user.
  - The calcFact algorithm/method computes the factorial.

Exercise 4-b

- Now modify the main algorithm/method as follows:
  - Check to see if the user entered a negative number.
  - If a negative number is entered, print a message saying that a nonnegative number is required and ask the user to enter another number.
  - Keep prompting the user enters a nonnegative number, after which the factorial can be computed (by calling the problem solving algorithm/method) and printed.
  - Develop the new main algorithm using Word and Viso – and insert into the file Lab4Ex4.doc.
  - Translate to Java. Start with Lab4Ex4a.java and create a new class Lab4Ex4b in the file Lab4Ex4b.java.