# Algorithms and Pseudo-Code
# Lecture 4

COEN 10

## Programming Fundamentals

★ What is a program?

1. Receive an input
2. Manipulate the input and generate a result
3. Output the result

## Programming Fundamentals

★ Two main concepts

- ◎ Need to handle input, output, and intermediary values
  - ✧ Variables
- ◎ Need to execute instructions to manipulate the input and output
  - ✧ Statements

## Programming Fundamentals

★ Variables

- ◎ Store data values
- ◎ Have a unique name
- ◎ Can be assigned different values during execution

## Programming Fundamentals

★Variables

```
input in
x = in
x = x * 2
output x
```

## Programming Fundamentals

★Statements

- ◎Basic Statements
  - ✧Simple instructions to the computer
- ◎Flow Statements
  - ✧Form a flow of execution

## Programming Fundamentals

★ Basic Statements

- ◎Assign values to variables
  - ✧Use the = character
  - ✧The value on the right is assigned to the variable on the left
- ◎Invoke procedures
  - ✧For example, to interact with the operating system

## Programming Fundamentals

★Example

```
input in
x = in
x = x * 2
output x
```

these lines represent function calls, in which the operating system provide the program with an input value and outputs the value provided by the program, respectively

these lines represent an assignment, in which the result of the multiplication is assigned to variable out

## Programming Fundamentals

flow statements determine the flow of execution that a computer follows to execute a task, i.e., to transform a set of input values into a sert of output values

★ Flow Statements

◎Sequential Statements

◎Conditional Statements

◎Repetition Statements

✧Counting

✧Conditional

◎Concurrent Statements

## Programming Fundamentals

★ Program

◎ Combination of basic and flow statements, sometimes nested

four basic elements of programming
穝equentiality:instructions are executed in sequence
穌ecision making: according to some condition, differents parths may be taken
穜epetition: instructions are executed in a loop
穋oncurrency: instructions are executed concurrently

## Programming Fundamentals

◎ Programming -- two steps

✧ Algorithm design

– Pseudo-code

a high-level language, which can then be coded, or translated into any programming language

✧ Coding

– Programming language

## Pseudo-Code

★ Sequential Statements

◎ The flow follows the list of instructions, one at a time

```
input z
x = 4
y = z
a = x * x
b = y * y * y
c = a / b
output c
```

## Pseudo-Code

★ Conditional Statements

◎ Enable the flow to execute different sets of instructions depending on a condition

```
x = input
if x is even
   x = x / 2
end if
y = x * 3
```

conditional statement

## Pseudo-Code

★ Two statements

```
x = input
y = input
if x is even
   x = x / 2
   y = y + 1
end if
y = y + x * 3
```

conditional statement

## Pseudo-Code

★ Two options

The first set of statements is executed if the condition is true. If the condition is not true, the alternative set of statements will be executed. Note that either the first or the alternative set executes.

```
x = input
if x is even
   x = x / 2
else
   x = x - 1
end if
y = x * 3
```

conditional statement

## Pseudo-Code

★ More than 2 options

One of the sequences executes depending on the conditions specified. The conditions are going to be checked in order.

```
x = input
y = input
if x is even
   x = x / 2
else if y is even
   x = x - 1
else
   y = y + 1
end if
y = y + x * 3
```

conditional statement

## Pseudo-Code

★ Repetition

◎ Enables the flow to execute a set of instructions

✧ For a number of times

✧ While a condition is true

A counting loop will assign an initial value to a variable and increment (or decrement) this value in each iteration of the loop, until the final value is reached. Note that the value of the variable change in each iteration of the loop, in which the actions specified in the loop are executed.

## Pseudo-Code

★Repetition Statement

◎Loop for a number of times

```
x = input
for i = 1 to 5
  x = x + i
end for
```

repetition statement

## Pseudo-Code

★One more example

```
x = input
y = input
for i = 1 to 5
  x = x + i
  y = y - i
end for
```

repetition statement

## Pseudo-Code

★ Repetition

◎ Loop while a condition is true

```
x = input
while x < 10
   x = x * 2
end while
```

repetition statement

## Pseudo-Code

★ One more example

```
x = input
y = input
while x < 100
   x = x * 2
   y = y - 1
end while
```

repetition statement

## Pseudo-Code

★ Repetition

◎ Loop while a condition is true, but execute at least once

```
x = input
do
   x = x * 2
while x < 10
```

repetition statement

## Pseudo-Code

★ Concurrent

◎ Execute one or more sets of instructions at the same time

```
do together
   { x = x + 1 }
   { y = y * 2 }
   { z = z / 5 }
end
```

concurrent statement

## Pseudo-Code

★ Examples

◎ Given two numbers, x and y, write the pseudo-code to output the greater one.

```
x = input
y = input
if x > y
output x
else
output y
end if
```

## Pseudo-Code

★Examples

◎Given two numbers, x and y, write the pseudo-code to output the greater one, or "same", if they are equal.

```
x = input
y = input
while x ≠ y
if x > y
output x
else
output y
end if
else
output "same"
end while
```

## Pseudo-Code

★Examples

◎Assuming a sequence of integer numbers, from x to y, where x < y, write the pseudo-code to output all the numbers in the sequence.

```
x = input
y = input
for i = x to y
output i
end for
```

## Pseudo-Code

★Examples

◎Assuming a sequence of integer numbers, from x to y, where x < y, write the pseudo-code to ouput all the even numbers in the sequence.

```
x = input
y = input
for i = x to y
if i %= 2 == ∅
output "yes"
else
output "no"
end if
```

## Pseudo-Code

★Examples

◎Assuming a sequence of integer numbers, from x to y, where x < y, write the pseudo-code to count how many of the numbers are multiple of 3.

```
x = input
y = input
for i = x to y
if i %= 3 == ∅
output i
end if
end for
```