COMP230: INTRODUCTION TO COMPUTERS AND PROGRAMMING

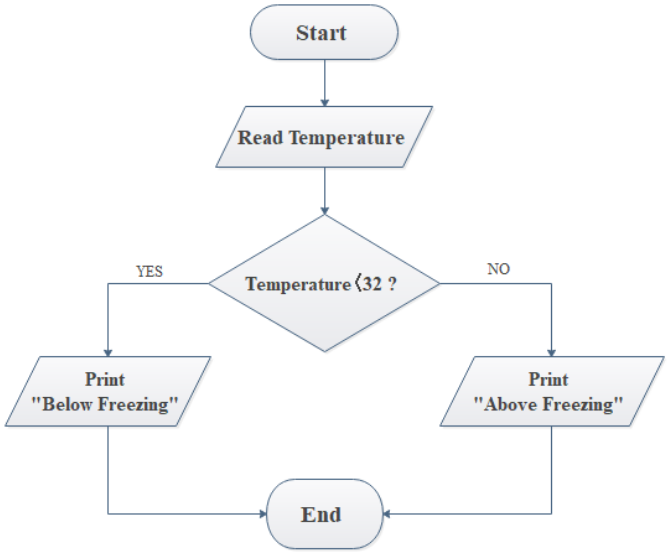
Algorithms

Dr. Radi Jarrar
Department of Computer Science
Birzeit University

Algorithm & Pseudocode

- An **algorithm** is a sequence of computational steps that transform input into output.
- Many ways to express algorithms
  - Pseudocode
  - Flowchart
- Pseudocode is a description of a computer programming algorithm that uses that structural conventions of programming languages, but omits language-specific syntax.

Flowchart



Example

- Let's say that you have a friend arriving at the airport, and your friend needs to get from the airport to your house. Here are three different algorithms that you might give your friend for getting to your home:

Example (2)

- The taxi algorithm:
  - Go to the taxi stand.
  - Get in a taxi.
  - Give the driver my address.

Example (3)

- The call-me algorithm:
  - When your plane arrives, call my cell phone.
  - Meet me outside baggage claim.

Example (4)

- The bus algorithm:
  - Outside baggage claim, catch bus number 70.
  - Transfer to bus 14 on Rukab Street.
  - Get off on Jerusalem street.
  - Walk two blocks north to my house.

Common Action Keywords

- Input: READ , OBTAIN, GET
- Output: PRINT, DISPLAY, SHOW
- Compute: COMPUTE, CALCULATE
- Initialize: SET
- Add one: INCREMENT.

Common Action Keywords (2)

- Input operations
  - To receive data values from the user.
  - Example
  - Get a value for r, the radius of the circle
- Output operations
  - To send results to the screen for display.
  - Example
  - Print the value of Area

Variables

- Variable
  - Named storage location that can hold a data value
- For example, Set x to 1. This will reserve a location in the memory and stores 1 in it.
- Let counter equals 10. This means reserve a memory location named counter and store 10 in it.

Types of Algorithm Logic

- Sequence Logic
- Selection Logic
- Iteration Logic

Sequence Logic

- Is used to perform instructions in a sequence (i.e., one after another).
- Instructions are written in the order in which they are to be performed.

Sequence Logic (4)

- Example: Write an algorithm to find and print the sum of two integers.

1. Ask user to enter first integer
2. Read the integer and save as integer_1
3. Ask user to enter the second integer
4. Read second integer and save as integer_2
5. Add integer_1 to integer_2 and save result as sum
6. Print sum to screen

Sequence Logic (5)

- Example: Write an algorithm to find and print the area of rectangle.

1. Ask user to enter the height of rectangle.
2. Read height and save as rectangle_height.
3. Ask user to enter the width of rectangle.
4. Read width and save as rectangle_width.
5. Multiply rectangle_heigh by rectangle_width and save the result as area.
6. Display area.

Sequence Logic (6)

- Example: Write an algorithm that computes the final price of an item after figuring its sales tax.

1. Ask user to enter a price
2. Read price and save it as item_price
3. Ask user to enter sales tax rate
4. Read sales tax rate and save it as tax_rate
5. Sales tax equals item_price multiplied by tax_rate
6. Final price equals price of item_price sales tax
7. Display Final price

Sequence Logic (7)

- Example: Write an algorithm to reverse any two digits number.

Sequence Logic (7)

• Example: Write an algorithm to reverse any two digits number.

1. Ask user to enter two digits number.
2. Read number and save as num.
3. Divide num by ten and save result as tens.
4. Divide num by ten and save remainder as rem.
5. Multiply rem by ten and save the result as rev.
6. Add tens to rev.
7. Print rev.

ALGORITHMS

Selection Logic

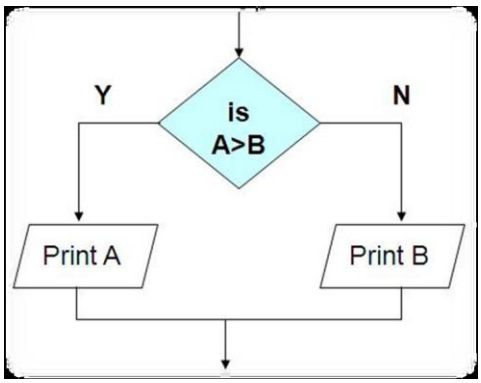
Selection Logic

- Also known as decision logic.
- Used to make decisions and for selecting the proper path out of two or more alternative paths in program logic.
- Selection logic is depicted as either an IF…THEN or and IF…THEN…ELSE structure.
- Ask questions and choose alternative actions based on the answers.

Selection Logic (2)

- Example IF A is greater than B THEN

```
        Print A
ELSE
        Print B
END IF
```



Selection Logic (3)

- The ELSE keyword is optional

IF condition THEN
  SEQUENCE
END
	IF condition THEN
	  SEQUENCE1
	ELSE
	  SEQUENCE2
	END IF
		IF condition1 THEN
		  SEQUENCE1
		ELSE IF condition2 THEN
		  SEQUENCE2
		ELSE IF condition3 THEN
		  SEQUENCE3
		ELSE IF condition4 THEN
		  SEQUENCE4
		END IF

Selection Logic – Logical & Relational Operators

Logical operators	Relational operators
• AND • OR	• Greater than • Greater than or equal • Less than • Less than or equal • Equal • Not equal

Selection Logic (4)

- Example: Write an algorithm to print passed or failed based on the student grade.

Selection Logic (4)

- Example: Write an algorithm to print passed or failed based on the student grade.

```
Ask user to enter student grade
Read grade and save as student_grade
IF student_grade greater than or equal sixty THEN
        Print "passed"
ELSE
        Print "failed"
END IF
```

Selection Logic (5)

- Example: Write an algorithm to find and print the maximum element of a set of 3 integers.

Selection Logic (5)

- Example: Write an algorithm to find and print the maximum element of a set of 3 integers.

1. Ask user to enter the first integer. 2. Read number and save as first_integer. 3. Ask user to enter the second integer. 4. Read number and save as second_integer. 5. Ask user to enter third integer. 6. Read number and save as third_integer.	7. Let max equal to the first_integer. 8. If max less than second_integer then set max to second_integer end if 9. If max less than third_integer then set max to third_integer end if 10. Print "the maximum integer is" max

Selection Logic (6)

- Example: Write an algorithm to find and print the smallest of three given numbers (assume all numbers are different).

Selection Logic (6)

- Example: Write an algorithm to find and print the smallest of three given numbers (assume all numbers are different).

1. Ask user to enter first number
2. Read number and save as num1
3. Ask user to enter second number
4. Read number and save as num2
5. Ask user to enter third number
6. Read number and save as num3
	7. IF num1 smaller than num2 **AND** num1 smaller than num3 THEN
	Print num1 is the smallest
	ELSE IF num2 smaller than num1 **AND** num2 smaller than num3 THEN
	Print num2 is the smallest
	ELSE
	Print num3 is the smallest
	END IF

Selection Logic (7)

- Example: Write an algorithm to read a number x and display its sign.

Selection Logic (7)

- Example: Write an algorithm to read a number x and display its sign.

1. Ask user to enter a number
2. Read number and save as X
3. IF x is greater than zero THEN
4. Print x “is positive”
5. ELSEIF x is equalzero THEN
6. Print x “is zero”
7. ELSE
8. Print x “is negative”
9. END IF

Selection Logic (8)

- Example: Write an algorithm that will input student average. If the average is greater than or equal to 60 and less than or equal to 70, the algorithm should display "Passed". If it is greater than 70 and less than or equal to 80, print "Good". If it is greater than 80 and less than 90, print "Very good". If it is greater than 90 , print "Excellent". If it is less than 60 the prints "Fail".

Selection Logic (8)

1. Ask user to enter student average
2. Read average and save as ag
3. IF ag is greater than or equal to sixty AND ag is less than or equal to seventy THEN
   Print "Pass"
   ELSE IF ag is greater than seventy AND ag is less than or equal to eighty THEN
   Print "Good"
   ELSE IF ag is greater than eighty AND ag is less than ninety THEN
   Print "Very good"
   ELSE IF ag is greater than ninety THEN
   Print "Excellent"
   ELSE
   Print "Fail"
   END IF

ALGORITHMS

Iterative Logic

Iterative Logic

- Used to produce loops when one or more instructions may be executed several times depending on some conditions.
- It uses three structures:

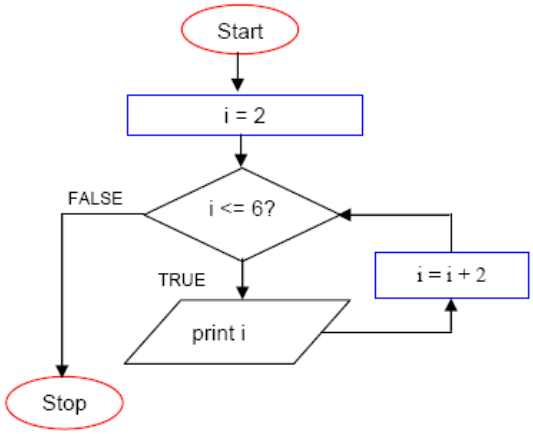
WHILE condition Sequence END WHILE	REPEAT Sequence UNTIL condition	FOR iteration bounds Sequence END FOR

Iterative Logic (2)

- Example:

Set i equal to two
    While i less than or equal six
        print i
        add two to i
    end while

Output:



Iterative Logic (3)

- Example: Write an algorithm to calculate the average of a set of 10 students. *Solution 1:*

Iterative Logic (3)

- Example: Write an algorithm to calculate the average of a set of 10 students. *Solution 1:*

Set counter to zero
Set total to zero
WHILE counter is less than ten
    Ask user to enter grade
    Read grade and save as gd
    Add the gd into the total
    increment counter
END WHILE
Set the average to the total divided by counter
Print "the average is " average

Iterative Logic (4)

- Example: Write an algorithm to calculate the average of a set of 10 students. *Solution 2:*

Set counter to one
Set total to zero
WHILE counter is less than or equal ten
    Ask user to enter grade
    Read grade and save as gd
    Add the gd into the total
    increment counter
END WHILE
Set the average to the total divided by counter
Print "the average is " average

Iterative Logic (5)

- Example: Write an algorithm to calculate the average of a set of n students. The user should enter -1 to stop.

Iterative Logic (5)

- Example: Write an algorithm to calculate the average of a set of n students. The user should enter -1 to stop.

```
Set counter to zero
Set total to zero
Ask user to enter grade
Input the first grade as gd
WHILE gd is not -1
        Add the gd into the total
        increment counter
        Ask user to enter grade
        Read grade and save as gd
END WHILE
```

```
IF counter is not equal to zero THEN
        Set average to the total divided by
counter
        Print average
ELSE
        Print "No grades were entered"
END IF
```

Iterative Logic (6)

- Example: Write an algorithm to print the number of passes and the number of failures in a set of n students. The user should enter -1 to stop.

```
Set counter to zero
Set passes to zero
Set failures to zero

Ask user to enter grade
Input the first grade as gd
WHILE gd is not -1
        IF gd is greater than or equal to 60
THEN
        Increment passes
ELSE
        Increment failures
END IF
        increment counter
        Ask user to enter grade
        Read grade and save as gd
END WHILE

Print "Number of passes is " passes
Print "Number of failures is " failures
```


Iterative Logic (7)

- Example: Write an algorithm that will count the number of student pass in a class and the amount failed. The pass mark is more than or equal to 65. Suppose the number of students is 30 . The algorithm should output the amount failed and passed.

Iterative Logic (7)

Set counter to zero
Set numberOfStudents to thirty
Set passCounterto zero
Set failureCounterto zero
WHILE counter less than numberOfStudents
    Ask user to enter student grade
    Read average and save as gd
    IF gdgreater than or equal sixty five THEN
        increment passCounter
    ELSE
        increment failureCounter
    END IF
    increment counter
END WHILE
Print "pass counter =" passCounter "and failure counter =" failureCounter

EXAMPLES ON ALGORITHMS

Example 1

- Write an algorithm that will accept the value of a side of a square and display its area where the formula is:
  area = side * side

Example 1

- Write an algorithm that will accept the value of a side of a square and display its area where the formula is:
  area = side * side

Ask the user to enter a side of a square

Read size and store it as side

Calculate Area as side multiplied by side

Print "The area is " Area

Example 2

- Write an algorithm that takes n integers and decides and prints the number of integers divisible by 3 and the number of integers not divisible by 3. Enter -99 to stop reading more integers.

Example 2

```
Set divisibleCounter to zero
Set notDivisibleCounter to zero
Ask the user to enter a number
Read number as X
WHILE X is not -99
        IF X is divisible by 3 THEN (or the remainder of dividing X by 3 is zero)
                Increment divisibleByCounter
        ELSE
                Increment notDivCounter
        ENDIF
        Ask the user to enter a number
        Read number as X
ENDWHILE
Print "The number of integers divisible by 3 is " divisibleCounter
Print "The number of integers not divisible by 3 is " notDivCounter
```

Example 3

- Write an algorithm to calculate & print the factorial of any number. E.g., if the user enters 5, then the algorithm should calculate the value of `5! = 5 * 4 * 3 * 2 * 1`

Example 3

- Write an algorithm to calculate & print the factorial of any number. E.g., if the user enters 5, then the algorithm should calculate the value of `5! = 5 * 4 * 3 * 2 * 1`

Set factorial to 1
Ask user to enter a number
Read number as N
FOR counter from 1 to N
    *factorial = factorial * counter*
    Increment counter
ENDFOR
Print factorial

Example 4

- Write an algorithm that checks if an input number is prime or not.

Example 4

- Write an algorithm that checks if an input number is prime or not.

```
Set isPrime to 1
Set counter to 2
Ask the user to input a number
Read number as N
WHILE counter is less than N
        IF N mod counter equals zero THEN
                Let isPrime equals zero
        ENDIF
ENDWHILE
IF isPrime equals 1 THEN
        Print N "is prime number"
ELSE
        Print N "is not prime number"
ENDIF
```

Example 5

- Write an algorithm that reads a number N and print all its divisors.

Example 5

- Write an algorithm that reads a number N and print all its divisors.

Set counter to 1
Ask the user to enter a number
Read number and save it as X
FOR counter from 1 to X
    IF the remainder of dividing X by counter is zero THEN
        Print counter "is a divisor of X"
    ENDIF
    Increment counter
ENDFOR

Example 6

- Write an algorithm that computes the sum of digits of a three-digit number.

Example 6

- Write an algorithm that computes the sum of digits of a three-digit number.

Ask the user to enter a number

Read the number and save it as X

Set Ones to the remainder of dividing X by 10

Set Hundreds to X divided by 100

Set Tens to X divided by 10 and then divided by 10

OR *Set Temp to X divided by 10*
Set Tens equals Temp divided by 10

Set sum equals Ones plus Tens plus Hundreds

Print sum

Example 7

- Write an algorithm that computes the sum of digits of a any input number.

Example 7

- Write an algorithm that computes the sum of digits of a any input number.

Set total to zero
Ask user to enter a number
Read the number and store it as X
WHILE X mod 10 Is not zero
    Let temp equals the remainder of dividing X by 10
    total equals total plus temp
    X equals X divided by 10
ENDWHILE