cs2010 lecture 2

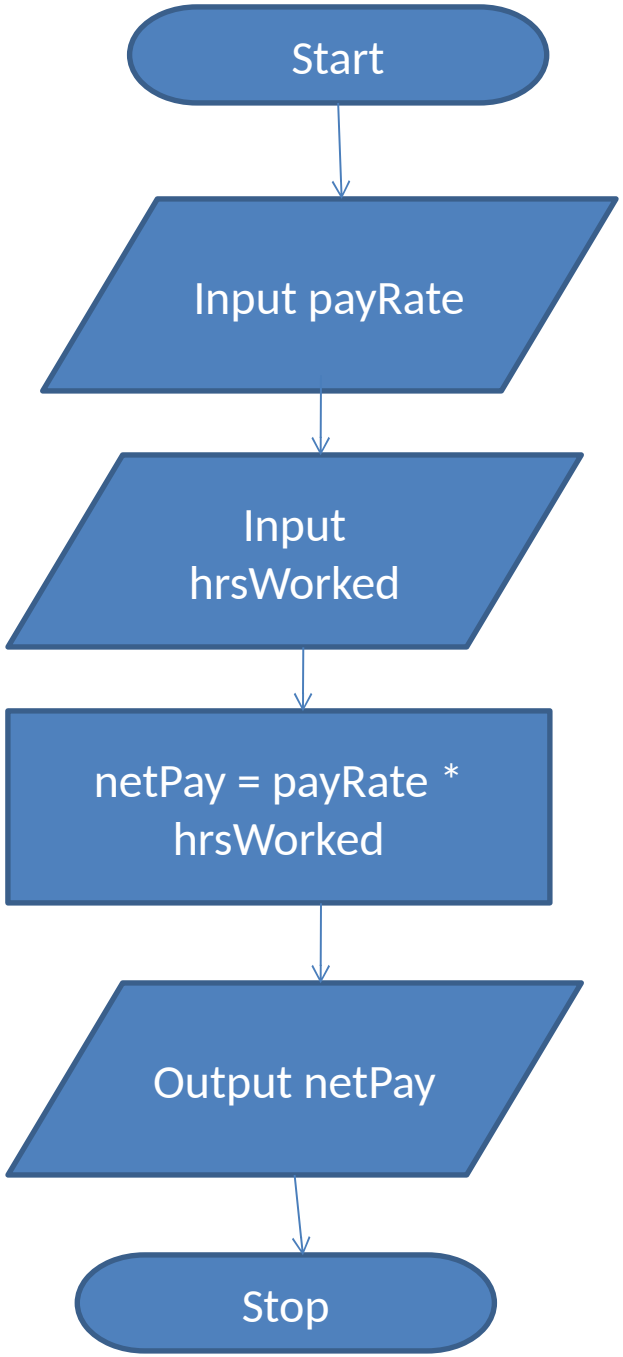
Based on slides from Prof. Sriharsha Mallapuram

Concepts:

- Code
- Pseudocode
- Flowcharts

Design tools

Pseudocode	**Flowchart**
•Can be done easily on word processor •Implements structured •design elements well •Not visual •No standard	•Standardized •visual •Difficult to modify •requires special software to do on computer



1. Write a program to calculate your paycheck. Input your hours worked and your pay rate and calculate net pay.

Pseudocode:

Input payRate
Input hrsWorked
netPay = payRate * hrsWorked
Output netPay

Step 1:Analysis
Define the Problem

IPO

1. Input: Determine what is given
   =>nouns, adjectives
2. Output: Determine what is required
   =>nouns, adjectives
3. Processing: Determine the transformations needed, actions needed to produce the required output
   =>verbs, adverbs

variables

- Input and output data
- Nouns
- Naming convention
  - No spaces
  - Meaningful names
  - Usually begin with lowercase
  - Examples: roomWidth, numPeople, studentName

Example 1

Read three numbers, add them together and print the total.

Input	Processing	Output

Variables only

1. Identify inputs
Underline nouns

Read three numbers, add them together and print the total.

- Break up several values into separate variables

Input	Processing	Output

1. Identify inputs
fill Input column first

Read three numbers, add them together and print the total.

- Break up several values into separate variables

Input	**Processing**	**Output**
num1		
num2		
num3		

2. Identify outputs
output column next

Read three numbers, add them together and print the total.

- Break up several values into separate variables
- No verbs here

Input	**Processing**	**Output**
num1		total
num2		
num3		

3. Processing-Define processing steps by bolding verbs

Read three numbers, **add** them together and **print** the total.

The read and print verbs are implicit to input and output, so we just need to add.

Input	**Processing**	**Output**
num1		total
num2		
num3		

Each verb is a process step

Read three numbers, **add** them together and **print** the total.

Input	Processing	Output
num1		total
num2		
num3		

Step 3: define list of actions

Read three numbers, **add** them together and **print** the total.

- *Hint*: Use verbs
- these steps usually involve the input and output defined in step 1

Input	Processing	Output
num1	Read input(s).	total
num2	**Add** numbers together	
num3	Print output(s).	

Example 2:

Write a program to prompt the operator for the maximum and minimum temperature readings on a particular day, accept those readings as integers, and calculate and display on the screen the average temperature.

Input	Processing	Output

Example 2:

Write a program to **prompt** the operator for the maximum and minimum temperature readings on a particular day, accept those readings as integers, and **calculate** and **display** on the screen the average temperature.

Input	Processing	Output

Example 2:

Write a program to **prompt** the operator for the maximum and minimum temperature readings on a particular day, accept those readings as integers, and **calculate** and **display** on the screen the average temperature.

Input	Processing	Output
maxTemp		avgTemp
minTemp		

Example 2:

Write a program to **prompt** the operator for the maximum and minimum temperature readings on a particular day, accept those readings as integers, and **calculate** and **display** on the screen the average temperature.

Input	Processing	Output
maxTemp	Read input(s).	avgTemp
minTemp		
	Print output(s).	

Example 2:

Write a program to prompt the operator for the maximum and minimum temperature readings on a particular day, accept those readings as integers, and **calculate** and display on the screen the average temperature.

Input	Processing	Output
maxTemp	Read input(s).	avgTemp
minTemp	Calculate average.	
	Print output(s).	

Example 2:

Write a program to prompt the operator for the maximum and minimum temperature readings on a particular day, accept those readings as integers, and **calculate** and display on the screen the average temperature.

Input	Processing	Output
maxTemp	Read input(s).	avgTemp
minTemp	Set avgTemp to be (minTemp + maxTemp / 2)	
	Print output(s).	

Flowchart Symbols

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectagle represents a process
	Decision	A diamond indicates a decision

Avg Temp program

- Have IPO table
- Write pseudocode
- Write flowchart
- Write code

Representing Algorithms (2 of 5)

- Sequential operations perform a single task
- The three basic sequential operations:
  - **Computation**: a single numeric calculation
  - **Input**: gets data values from outside the algorithm
  - **Output**: sends data values to the outside world
- A **sequential algorithm** is made up only of sequential operations
- A **variable** is a named storage location to hold a data value
- Example: computing average miles per gallon

Representing Algorithms Sequential Algorithm

FIGURE 2.3

Step	Operation
1	Get values for *gallons used, starting mileage, ending mileage*
2	Set value of *distance driven* to *(ending mileage - starting mileage)*
3	Set value of *average miles per gallon* to *(distance driven ÷ gallons used)*
4	Print the value of *average miles per gallon*
5	Stop

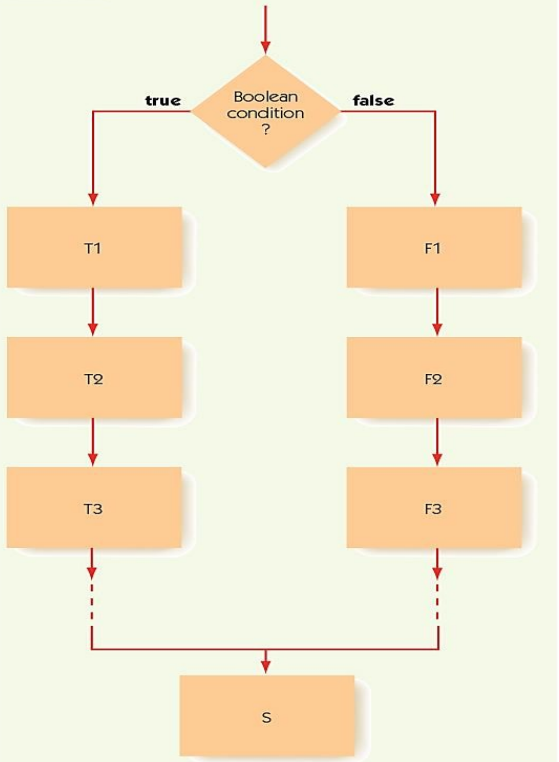
Algorithm for computing average miles per gallon (version 1)

Representing Algorithms (3 of 5)

- **Control operation:** changes the normal flow of control
- **Conditional statement**: asks a question and selects among alternative options:
    1. Evaluate the true/false condition
    2. If the condition is true, then do the first set of operations and skip the second set
    3. If the condition is false, skip the first set of operations and do the second set
- Example: check for good or bad gas mileage

Representing Algorithms Conditional Statement (1 of 2)

FIGURE 2.4



The if/then/else pseudocode statement

Representing Algorithms Conditional Statement (2 of 2)

FIGURE 2.5

Step	Operation
1	Get values for *gallons used, starting mileage, ending mileage*
2	Set value of *distance driven* to (*ending mileage – starting mileage*)
3	Set value of *average miles per gallon* to (*distance driven ÷ gallons used*)
4	Print the value of *average miles per gallon*
5	If *average miles per gallon* is >25.0 then
6	Print the message 'You are getting good gas mileage'
	Else
7	Print the message 'You are NOT getting good gas mileage'
8	Stop

Second version of the average miles per gallon algorithm

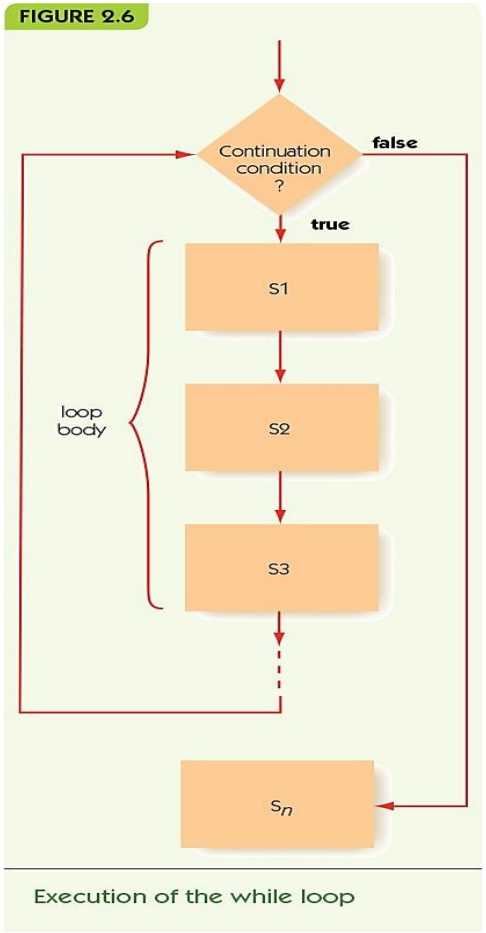
Good mileage program

- Have pseudocode table
- Write IPO table
- Write flowchart
- Write code

Representing Algorithms (4 of 5)

- **Iteration:** an operation that causes looping, repeating a block of instructions
- While statement repeats while a condition remains true
  - **Continuation condition**: a test to see if while loop should continue
  - **Loop body**: instructions to perform repeatedly
- Example: repeated mileage calculations

Representing Algorithms Iteration and Loop Body (1 of 2)



Representing Algorithms Iteration and Loop Body (2 of 2)

FIGURE 2.7

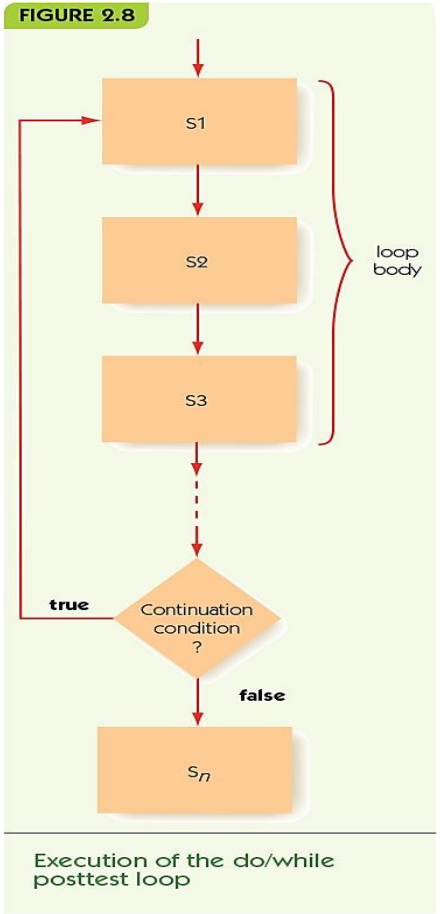
Step	Operation
1	*response* = Yes
2	While (*response* = Yes) do Steps 3 through 11
3	Get values for *gallons used*, *starting mileage*, *ending mileage*
4	Set value of *distance driven* to (*ending mileage* – *starting mileage*)
5	Set value of *average miles per gallon* to (*distance driven* ÷ *gallons used*)
6	Print the value of *average miles per gallon*
7	If average miles per gallon > 25.0 then
8	Print the message 'You are getting good gas mileage'
	Else
9	Print the message 'You are NOT getting good gas mileage'
10	Print the message 'Do you want to do this again? Enter Yes or No'
11	Get a new value for *response* from the user
12	Stop

Third version of the average miles per gallon algorithm

Representing Algorithms (5 of 5)

- Do/while, alternate iterative operation
  - Continuation condition appears at the end
  - Loop body always performed at least once
- **Primitive operations:** sequential, conditional, and iterative are all that is needed

Representing Algorithms Do/While Posttest Loop (1 of 2)



Representing Algorithms Do/While Posttest Loop (2 of 2)

FIGURE 2.9

Computation:
Set the value of "variable" to "arithmetic expression"

Input/Output:
Get a value for "variable", "variable"...
Print the value of "variable", "variable", ...
Print the message 'message'

Conditional:
If "a true/false condition" is true then
first set of algorithmic operations
Else
second set of algorithmic operations

Iterative:
While ("a true/false condition") do Step *i* through Step *j*
Step *i*: operation
.
.
.
Step *j*: operation

While ("a true/false condition") do
operation
.
.
.
operation
End of the loop

Do
operation
operation
.
.
.
operation
While ("a true/false condition")

Summary of pseudocode language instructions

Examples of Algorithmic Problem Solving Example 1: Go Forth and Multiply (1 of 5)

Given two nonnegative integer values, $a \geq 0$, $b \geq 0$, compute and output the product ($a \times b$) using the technique of repeated addition. That is, determine the value of the sum $a + a + a + \ldots + a$ (b times).

Examples of Algorithmic Problem Solving Example 1: Go Forth and Multiply (2 of 5)

- Get input values
  - Get values for *a* and *b*
- Compute the answer
  - Loop *b* times, adding each time*
- Output the result
  - Print the final value*

* steps need elaboration

Examples of Algorithmic Problem Solving Example 1: Go Forth and Multiply (3 of 5)

- Loop b times, adding each time
  - Get values for *a* and *b*
  - Set the value of *count* to 0
  - While (*count* < *b*) do
    - … the rest of the loop*
    - Set the value of *count* to (*count* + 1)
  - End of the loop

* steps need elaboration

Examples of Algorithmic Problem Solving Example 1: Go Forth and Multiply (4 of 5)

- Loop b times, adding each time
  - Get values for *a* and *b*
  - Set the value of *count* to 0
  - Set the value of *product* to 0
  - While (*count* < *b*) do
    - Set the value of *product* to (*product* + *a*)
    - Set the value of *count* to (*count* + 1)
  - End of the loop
- Output the result
  - Print the value of *product*

Examples of Algorithmic Problem Solving Example 1: Go Forth and Multiply (5 of 5)

FIGURE 2.10

```
Get values for a and b
If (either a = 0 or b = 0) then
    Set the value of product to 0
Else
    Set the value of count to 0
    Set the value of product to 0
    While (count < b) do
        Set the value of product to (product + a)
        Set the value of count to (count + 1)
    End of loop
Print the value of product
Stop
```

Algorithm for multiplication of nonnegative values via repeated addition

Alternative representations for that pseudocode

IPO Table?

- Inputs
- Whole algo is process
- Outputs

Flow chart

- Let’s draw it

Code?

- Let’s write it

Flowchart Symbols

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectagle represents a process
	Decision	A diamond indicates a decision