LECTURE 05
PROGRAM DESIGN

PSEUDOCODE

Overview

- Flowcharts were the first design tool to be widely used.
- Drawback - it does not reflect some of the concepts of structured programming very well.

  eg: modularity
- Pseudocode, is a popular tool and has features that make it more reflective of the structured concepts.
- Drawback - is that the narrative presentation is not as easy to understand and/or follow.

Pseudocode

A way of *expressing* algorithms that uses a mixture of *English phrases* and *indention* to make the steps in the solution explicit.

Characteristics of Pseudocode

- Statements use simple structured English
- Write only one statement per line
- Indent to show hierarchy
- Each set of instructions is written from top to bottom
- “End” multiline Structure
- A block of statements may be formed into modules. That group will be given a name.

Pseudocode Functionality

Common operations done by a computer are as follows:

1. Receiving input
2. Put out information
3. Perform arithmetic operations
4. Assigning value to a variable or memory location
5. Based on a condition to select one or two alternative actions
6. Repeat a group of actions

1. Receiving input

When computer is required to receive information or input from a source.

Get
- used when the algorithm is to receive input from the keyboard / system.

Read
- algorithm receives the input from a record on a file

Eg: Get emp_number

Get system time

Read num1, num2

2. Put out information

When computer is required to supply information or output to a device.

Print	• output is to be sent to the printer
Write	• used when the output is to be written to a file
Put, Output, Display	• used when the output is to be written to the screen
Prompt	• required before an input instruction Get. Sends a message to the user in getting the input

Eg: Print “Processing records are completed”

write student record to master file

write emp_num, address, pay to employee file

Put name, address , postcode

Output student_grade

Display class_average

prompt for student_grade

3. Perform arithmetic operations

Compute, Calculate

- Performing mathematical calculations or applying formula

Mathematical symbol or word may be used.

+ , -, *, /, ()

Discuss : Order of Operations

Eg: Add student_mark to class_total

count = count + 1

C = (F – 32) *5/9

Compute C = (F-32)*5/9

4. Assigning value to a variable or memory location

Set, Initialize	• To give an initial value to the data
= , ←	• assign a value as a result of some processing
Save, Store	• To keep a variable for later use

Eg:

Initialize count to zero

Set class_total to 0

monthly_pay = basic_sal + overtime

monthly_pay ← basic_sal + overtime

Store customer_num in last_customer_num

5. Based on a condition to select one or two alternative actions

Eg:

```
IF std_gender = "female" THEN
        Add 1 to total_females
ELSE
        Add 1 to total_males
ENDIF
```

6. Repeat a group of actions

DOWHILE ENDDO

Eg:

```
DOWHILE count <15
        Get temp_F
        temp_C = (temp_F - 32)*5/9
        Display temp_C
ENDDO
```

Defining Variable Names

In the Solution Algorithm, the programmer should ensure that naming is done in the following manner:

- Use meaningful names for the variables or objects in the problem

  eg: number1, number2, number3 more meaningful than A, B, C

- Used word separator if more than one word

  eg: sales_tax, word_count

- Or Capital letter as separator

  eg: salesTax, wordCount

The Structure Theorem

In a Pseudocode, the statements can be represented using the following three structures.

- Sequence
- Selection
- Repetition

1. Sequence

Is the straightforward execution of one processing step after another.

```
Statement 1
Statement 2
Statement 3
```

Eg:

```
Read unit_price
Read qty
total_price = unit_price * qty
Display total_price
```

2. Selection

Evaluation of the condition a choice between two actions.

```
IF condition is true THEN
        Statement block in true case
ELSE
        Statement block for false case
ENDIF
```

Eg:

```
Get student_mark
IF student_mark>50 THEN
        Display "Student is Passed"
ELSE
        Display "Student is Failed"
ENDIF
```

Selection Control Structures

- Simple Selection
- NULL else statement
- Combined IF statement
- Nested IF statement
- CASE statement

3. Repetition

The presentation of a set of instructions to be performed repeatedly, as long as a condition is true

```
DOWHILE condition is true
        Statement block
ENDDO
```

Eg:

```
Set count to zero
Set total_number to zero
DOWHILE count<10
        Get number
        Add number to total_number
        Add 1 to count
ENDDO
```

Repetition Control Structures

- DOWHILE/ENDDO looping [leading decision loop]
- REPEAT/UNTIL looping [trailing decision loop]
- Counted loop

Guidelines on Writing the Pseudocode

- A name should be given to the Pseudocode, which describes the function.
- An END statement used to indicate the Pseudocode is complete.
- All processing steps between the Pseudocode name and END statement should be indented for readability.
- Each processing step in the defining diagram relates directly to one or more statements in the Pseudocode.

Example 1 – Temperature Conversion

A program is to be written that will accept a Fahrenheit temperature, convert it to Celsius and display the converted temperature to the screen.

- Defining Diagram

Input	Processing	Output
f_temp	Accept f-temp Calculate c_temp Display c_temp	c_temp

- Solution Algorithm

```
        Convert_temperature
1               Prompt for f_temp
2               Get f_temp
3               c_temp = (f_temp – 32) * 5/9
4               Display c_temp
        END
```

Checking the Pseudocode

Tracing through the logic of the algorithm with some chosen data.

Desk Checking Steps

1. Choose valid simple input test case (2-3 enough)
2. Establish what the expected result should be.
3. Make a table of relevant variable names & output
4. Checking the test case line by line, step by step,
5. Repeat process 4 for other test case
6. Check if expected result 2 matches with actual result 5

Desk Checking for Example 1

Input Data

Test Data Set	#1	#2
f_temp	32	50

Expected Results

	#1	#2
c_temp	0	10

Note : For simplicity only the changed value against the statement can be highlighted in the table

Test Data Set#	Statement No	f_temp	c_temp	Output
#1	1,2	32		
	3	32	0	
	4	32	0	0
#2	1,2	50		
	3	50	10	
	4	50	10	10

Since the expected results and the actual results match, the Pseudocode is correct.

Example 2 – Validate User

Write a Pseudocode for a program that allows the user to input a username and a password. The username and passwords are compared with an existing username (Mark) and password (AB55TR).

If they match then print "PASS" on the screen.

If the words do not match "NO ACCESS" is shown.

Solution Algorithm

```
Validate_User
1     Prompt for username, password
2     Get username, password
3     IF (username=="Mark" AND password=="AB55TR") THEN
4           message = "PASS"
5      ELSE
6           message = "NO ACCESS"
      ENDIF
7     Display message
END
```

Note : The IF…ELSE….ENDIF is processed as a single Algorithm step

Input Data

Test Data Set	#1	#2
Username	Mark	Mark
password	AB55TR	ABCDEF

Expected Results

Test Data Set	#1	#2
Message	PASS	NO ACCESS

Test Data Set#	Statement No	Username	password	Message	Output
#1	1,2	Mark	AB55TR		
	3	Mark	AB55TR	PASS	
	4	Mark	AB55TR	PASS	
	7	Mark	AB55TR	PASS	PASS
#2	1,2	Mark	ABCDEF		
	3	Mark	ABCDEF	NO ACCESS	
	4	Mark	ABCDEF	NO ACCESS	
	7	Mark	ABCDEF	NO ACCESS	NO ACCESS

Expected outcome matches the actual results.

Example 3 – Calculate Total

Write a Pseudocode to read 20 numbers and to display their total.

Solution Algorithm

```
        Calculate_total
1               Set count to zero
2               Set total to zero
3               DOWHILE count <20
4                       Prompt for number
5                       Get number
6                       Add number to total
7                       count = count +1
                ENDDO
8               Display total
        END
```

Input Data

Test Data Set #1

Input#	1	2
number	50	40

Expected Results

total = 90 (after 2 iterations)

Test Data Set#	Statement No	Count	DOWHILE condition	number	total	Output
#1	1	0				
	2	0			0	
	3	0	TRUE		0	
	4,5	0		50	0	
	6	0		50	50	
	7	1		50	50	
	3	1	TRUE	50	50	
	4,5	1		40	50	
	6	1		40	90	
	7	2		40	90	

Expected outcome matches the actual results.