

Programming Fundamentals

Pseudocodes

Week 3 | Prasan Yapa

Learning Outcomes

- Covers part of LO1 & LO2 for Module
- On completion of this lecture, students are expected to be able to:
 - Identify the use of pseudocodes in program design.
 - Demonstrate competence in designing pseudocodes in program design to solve problems.

What Is Pseudo-Code?

- **Pseudo code** is an outline of a program written in a way that it can be easily converted into a computer programming language
- It mixes natural language with standard programming language constructs, such as
 - Expressions: $c = a + b$
 - Method Declarations: Algorithm name (param1,param2,.....)
 - Decision Structures: if condition then true-actions [else false-actions].
 - While-Loops: While condition do actions. We use indentation to indicate what actions should be included in the loop actions.
 - Array Indexing: $A[i]$ represents the i th cell in the array A . The cells of an n -celled array A are indexed from $A[0]$ to $A[n-1]$.

How to Write Pseudo-code..

- The Six Basic Computer Operations
 1. A computer can receive information
 2. A computer can out put information
 3. A computer can perform arithmetic
 4. A computer can assign a value to a variable or memory location
 5. A computer can compare two variables and select one of two alternative actions
 6. A computer can repeat a group of actions

1. A Computer Can Receive Information

- When a computer is required to receive information or input from a particular source, whether it be a terminal, a disk or any other device, the verbs **Read**, **Input** and **Get** are used in pseudo-code.
 - Read student name
 - Get system date
 - Read number_1, number_2
 - Get tax-code
 - Input marks

2. A Computer Can Put Out Information

- When a computer is required to supply information or output to a device, the verbs **Print**, **Write**, **Put**, **Output** or **Display** are used in pseudo-code.
 - Print 'Program Completed'
 - Write customer record to master file
 - Output name, address and postcode
 - Output total-tax
 - Display 'End of data'

3. Computer Can Perform Arithmetic

- Computers were invented to perform arithmetic. To write a mathematical calculation or formula either actual mathematical symbols or the words for those symbols can be used.
 - Add number to total
 - $\text{total} = \text{total} + \text{number}$

Arithmetic Operators used in Pseudo-code

- Arithmetic operators
 - + Addition
 - - Subtraction
 - * Multiplication
 - / Division
 - = Assignment
- $()$, +, -, *, /

4. Computer Can Assign a Value to a Variable or Memory Location

- Computers can assign or change the value of a variable.
Some common command for assignment are SET, =, STORE, INITIALIZE
 - Initialize count to zero
 - Set student_count to 0
 - Total_price = cost_price + VAT
 - Store customer-num in lost-customer-num

5. Can Compare Two Variables and Select One of Two Alternative Actions

- Computers can compare two variables and then, as a result of the comparison, select one of the two alternative actions.
- To represent this operation in pseudo-code, special keywords are used: **IF**, **THEN**, and **ELSE**. If the question in the IF clause evaluates to True, the statements in the THEN path are executed. Otherwise the statements in the ELSE path are executed.

```

IF (age>18)      THEN
    display 'Adult'
ELSE
    display 'Child'
END IF
  
```

The IF ...THEN ...ELSE Construct

- A program which asks a user for 2 numbers (A, B) and calculates the value of A divided by B. However, if B is 0 a message is printed which says that division by 0 is not allowed.

Start

Integer A, B, C

Get A, B

IF (B ==0) THEN

 Display 'Division by 0 is not allowed'

ELSE

 C = A / B

 Display 'A=', A

 Display 'B=', B

 Display 'Answer=', C

END IF

STOP

The IF ...THEN ...ELSE Construct consists of:

- The word IF
- A condition (B = 0, in this example)
- The word THEN
- One or more statements called 'the THEN part' (1, in this example)
- One or more statements called 'the ELSE part' (2, in this example)
- The word, END IF indicating the end of the construct.

Also, the statements are indented so that we can see at a glance the structure of the construct, especially which statements belong to the THEN and ELSE parts.

6. Computer Can Repeat a Group of Actions

- When there is a sequence of processing steps, which need to be repeated, two special keywords, WHILE DO and END WHILE, are used in pseudo-code. The condition for the repetition of a group of actions is established in the WHILE DO clause, and the actions to be repeated are listed beneath it.
- Some common commands for repeat are:
 - FOR loop,
 - WHILE loop,
 - REPEAT UNTIL loop

The While Construct

- Get value for C. As long as C is not zero, convert C into F and get another value for C. When C is zero, stop.

Integer C, F

Get C

WHILE (C !=0)DO

$F = 32 + (9 * C / 5)$

 Display 'Centigrade=', C

 Display 'Fahrenheit =', F

 Get C

END WHILE

STOP

The While Construct

- A WHILE Loop consists of:
 - The word WHILE
 - A condition (C is not zero, in this example)
 - The word DO
 - One or more statements (3, in this example)
 - The word, END WHILE indicating the end of the loop

Also, the statements are indented so that we can see at a glance the structure of loop.

Exercise

- A user is asked to enter a set of positive numbers, one at a time. User enters 0 to indicate that he/she has no more numbers to enter. Develop a pseudo-code to print the largest number entered.



This program assumes that at least one number will be supplied before the 0 is entered. The first number is stored in the variable largest.

Start

Integer largest, newNumber

Get largest

Get newNumber

WHILE (newNumber != 0) DO

 IF (newNumber > largest) THEN

 largest = newNumber

 END IF

 Get newNumber

END WHILE

Display 'Largest number entered is ' largest

STOP

The Three Essential Programming Constructs

- Loop (The repetition construct)
 - IF ...THEN ...ELSE Construct (The selection construct)
 - The Sequence construct
-
- With these 3 constructs one can specify any algorithm without the use of 'go to' statements.
 - These constructs form the basis for structured programming.
 - Sometimes we use variations of these.

Meaningful names

- > Programmer must introduce some unique names, which will be used to represent the variables in the problem. All names should be meaningful.
- > Often a name describes the type of data stored in a particular variable.
- for example, number1, number2 and number3 are more meaningful names for three numbers than A, B and C.
- > When more than one word is used in the name of a variable, then underscores are useful as word separators, for example sales_tax and word_count
- > do not have a space in a variable name, as a space would signal the end of the variable name and thus imply that there were two variables.
- Example. incomeTax

Rules for Pseudo-code

- Write only one statement per line
- Capitalize initial keyword
- Indent to show hierarchy
- End multi-line structures
- Keep statements, language independent

Operators Used in Pseudo-code

- Relational operators
 - < Less than
 - > Greater than
 - <= Less than or Equal to
 - >= Greater than or Equal to
 - <> or != Not Equal to
 - == Equal to

Case 2

Write a program with which you can convert a human height given in feet and inches to centimeters. The program should ask the user to type in his or her height in two parts: first the height in feet and then the inches part for the height. (A person can say that his or her height is, for example, 5 feet and 9 inches. That would be $30.48 * 5 + 2.54 * 9$ centimeters.)

Case 3

A program is required to read and print a series of names and exam scores for students enrolled in a mathematics course. The class average is to be calculated and printed at the end of the report. Scores can range from 0 to 100. The last record contains a blank name and a score of 999 and is not to be included in the calculations.

Case 4

Write a program that prints a conversion table from miles to kilometers. The program should produce the following output to the screen

miles	kilometers
10.00	16.09
20.00	32.19
30.00	48.28
40.00	64.37
50.00	80.47
60.00	96.56
70.00	112.65
80.00	128.74
90.00	144.84
100.00	160.93
110.00	177.02

Case 4

Improve the program so that it prints, after the table created in the previous exercise, a table that contains conversions from kilometers to miles.

Improve your program so that the user can select what kind of conversion table must be printed. In the beginning your program should print the following text.

This program prints conversion tables.

Type a letter to select a conversion table

m miles to kilometers

k kilometers to miles

After these lines are printed your program should read one character from the keyboard. According to the character the program should print the correct conversion table.

Logical Operators Used in Pseudo-code

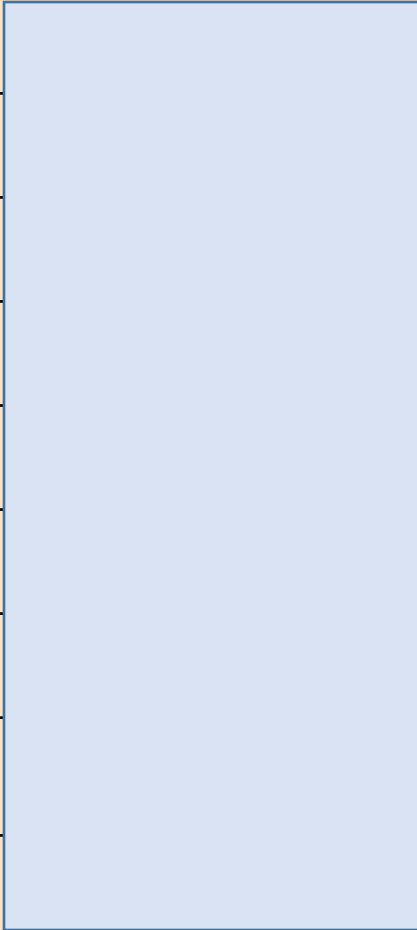
- AND
- OR
- NOT

AND - IF (x == 32) AND (y == 7) THEN sumxy = x + y

OR - IF (letter == 'A') OR (letter == 'E') THEN DISPLAY 'Vowel'

NOT - IF NOT (letter = 'A') THEN DISPLAY 'Not letter A'

Use of relational and logical operators, assume that A contains 20 and B contains 15

Expression	Result	
A >= 20		
A > 20		
A == B		
A == B + 5		
(A > B) AND (A > 20)		
(A > B) OR (B > A)		
(A < B) OR (B > A)		
NOT (A > B)		
NOT (NOT (A > B))		

Order of Precedence

- ()
- NOT
- *, /, AND
- +, -, OR
- ==, <, >, <>, <=, >=

*If a is 10 and b is 5 then, $x = a * (b + 7) + (a/b - 3)$?*

Summary

- The five basic operations of a program
- The three essential constructs in a program
 - The Selection construct
 - The Repetition construct
 - The Sequence construct
- Operators used in Pseudo-code

Thank You !!