Programming Logic & Techniques (PLT)

Objectives

- Learn to write a PSEUDOCODE
- Understand good programming techniques
- Learn how to analyze a problem
- Translate mathematical logic and analytical
- Skills into pseudo codes
- Realize the OPTIMIZED way to solve a problem

What is a Pseudocode?

- A formal representation of programming logic in a language independent manner

- Supports all the basic programming constructs like conditional statements and iterations

- Other such representations are
  - Flowchart
  - Algorithm

Structure of a Pseudocode

```
Start
        Declarations
                :
                :
        Statements
                :
                :
Stop
```

Structure

```
Start
        num1 ← 0(integer);
        num2 ← 0(integer);
        sum← 0(integer);

        Accept num1,num2;
        sum ← num1 + num2;
        Display sum;

Stop
```

Example

Some rules and syntax

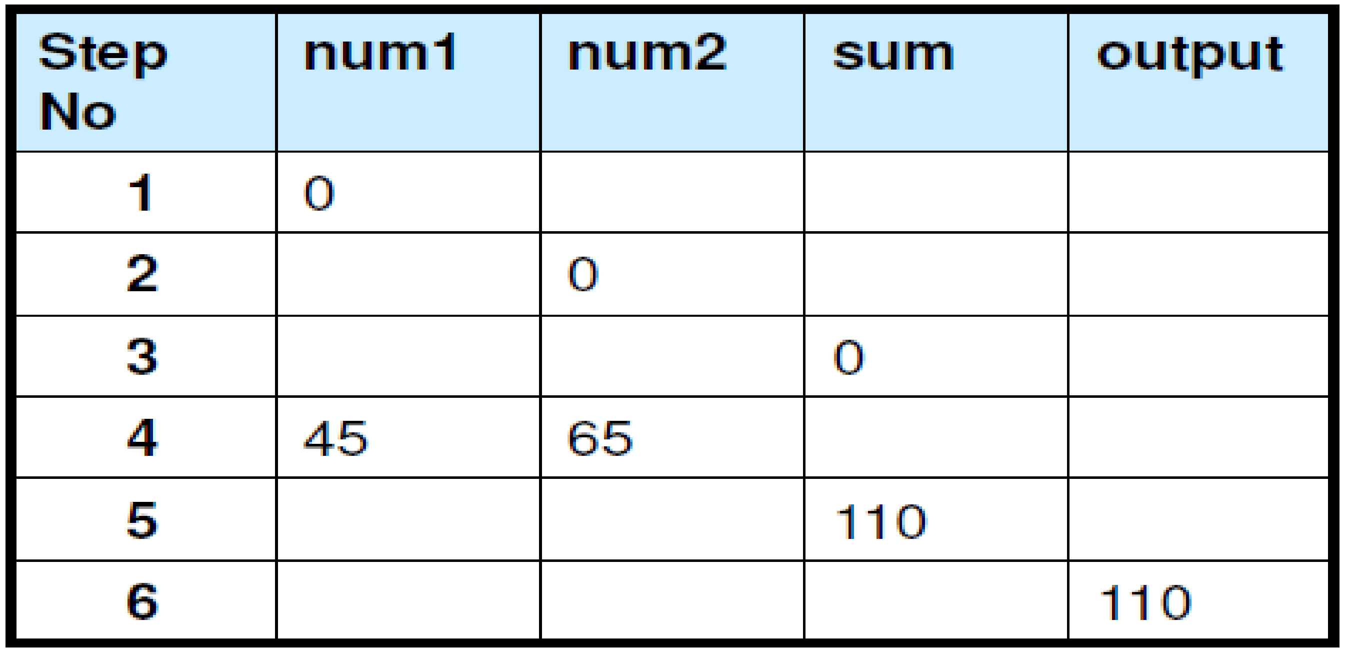
- Each variable declared is initialized with
- ‘some value or null’ for that data type
- The data types are
  - **Integer**
  - **Double**
  - **Char**
  - **String**
- Ex.
  - **age 0 (integer);**
  - **salary 0.0 (double);**
  - **Choice ‘y’ (char);**
  - **Name “Manhattan” (string);**
- Each statement is terminated with a semi-colon (;)

Follow the rules

- Writing a good code is not just solving the problem, but also complying with the syntax
- Use good naming style for the variables declared
- Indentation is mandatory and improves readability of the code
- Test your code by using the Dry Run technique
  - Ability to test your code is as critical as writing the code

Dry Run

- A simple dry run for the example pseudocode



Step No	num1	num2	sum	output
1	0			
2		0		
3			0	
4	45	65		
5			110	
6				110

Exercise

- Write a pseudocode to accept principle, rate of interest and time. Calculate simple interest and display the same

- Write a pseudocode to accept two numbers. Display the two numbers. Swap the two numbers and display them again.

Conditional statements

- The ‘if’ statement is used for conditional branching

General Format

```
    if (condition) then
    statement1;
    statement2;
    :
    else
    statement1;
    statement2;
    :
    endif;
```

Example

```
    if (amtWithdraw < balance) then

    balance  balance - amtWithdraw;

    else

    display “insufficient funds!”;

    endif;
```

Relational & Logical Operators

- Relational Operators
  - >
  - <
  - >=
  - <=
  - !=
  - ==
- Logical Operators
  - and
  - or
  - Not
- Arithmetic Operators
  - +, -, *, /, %, ^

Exercise

- Write a pseudocode to accept a number and display whether it is an even or odd number

- Write a pseudocode to accept a double value. Separate the whole value from the fractional value and store them in two variables. Display the same.

Nested if structures

- A completely nested (contained) if structure inside another if … else … structure
  - Allows multiple branching
- General Format

```
if (condition) then
    statement1;
    :
    if (condition) then
    statement2;
    else
    statement3;
    endif;
else
    statement1;
    if (condition) then
    statement2;
    else
    statement3;
    endif;
    :
endif;
```

Exercise

- Write a pseudocode to accept a student’s name and scores in three subject. Display the average and total. Display whether the student has secured 1st, 2nd, pass class or has failed. 1st class is for a score of 60 and above, 2nd is for a score of 50 and above , while pass class is for a score of 35 and above. If the score is less than 35, then the student fails.

- Write a pseudocode to find the largest and second largest of 3 numbers

Exercise

- Write a pseudocode to accept name, empId, basic , special allowances, percentage of bonus and monthly tax saving investments. The gross monthly salary is basic + special allowances. Compute the annual salary. The gross annual salary also includes the bonus. Compute the annual net salary, by deducting taxes as suggested.
  - Income upto 1 lac – exempted
  - Income from 1 to 1.5 lac – 20%
  - Income from 1.5 lac onwards – 30%

However if there is any tax saving investment, then there is further exemption of upto 1 lac annually.

This would mean that by having tax saving investments of about 1 lac, an income of 2 lacs is non-taxable.

Display the annual gross, annual net and tax payable.

Exercise

- A vendor offers software services to a client. Each resource is billed at some dollar rate per hour. The total cost of the project for the client is therefore, the total number of hours contributed by all the vendor resources * the dollar rate / hour. There are however some variants.
  - The vendor might have purchased hardware/infrastructure or software licenses needed for the project. The vendor might have utilized external consultants for the project.
  - The client looks at the vendor as a one-stop solution and hence external resources employed by the vendor need to be paid by the vendor.
  - It might however be possible that the vendor's hardware and software purchases are borne by the client. In this case, the client pays the vendor 30% of the hardware/infrastructure costs. In case of software licenses, the client pays the vendor 50% of the cost, if they are commonly available and used, or 100% if the software is infrequently used or is proprietary client technology.

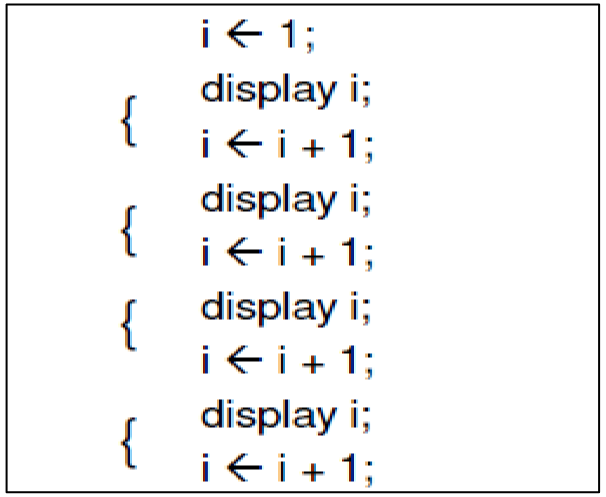
The external consultants employed by the vendor will come at a dollar rate per hour. Accept the suitable inputs and display the profits / loss realized by the vendor.

Structured Loops

- There are three types of structured loops
  - While
  - Repeat…Until (Do-While)
  - For
- Different languages support them in different ways
- Java / C / C++, C# does not support the Repeat Until loop
- Java and other language uses a different version of while as the post tested loop(Do-While)
- Pascal however supports the Repeat…Until loop

The need for iteration

- Consider the following code to display numbers from 1 to 10



```
        i ← 1;
        display i;
    {
        i ← i + 1;
        display i;
    {
        i ← i + 1;
        display i;
    {
        i ← i + 1;
        display i;
    {
        i ← i + 1;
```

- Notice how the same instructions are repeated. A simpler way of achieving the same is to request the system to repeat the
- same steps multiple times, than the programmer repeating the steps as many times.
- Imagine generating hundreds of numbers or finding the annual net salary of all the employees in an organization

The While Loop

- General Format

```
While (condition)
statement1;
statement2;
:
End While;
```

- Example

```
i← 1;
While (i <= 10)
   display i;
   i ← i + 1;
End While;
```

Requirements & Properties of the While loop

- **Requirements**
  - A proper initial value to the control variable(s)
  - A proper condition
  - A change of value in the body of the loop, to the
  - control variable(s)
  - Proper begin and end of the loop
- **Properties**
  - Pre-tested loop
    - Need not be executed once
  - Enters when ‘true’, exits when ‘false’
- The same requirements and properties apply for **NESTED LOOPS**

Exercise

- Write a pseudocode to find the sum of all odd numbers from 1 to N. Accept N. Display the sum.

- Write a pseudocode to find the reverse of a number. Store the reverse value in a different variable.Display the reverse.

- Write a pseudocode to display a number in words.
  - Ex. 270176
    - Output: Two Seven Zero One Seven Six

- **Write a DRY RUN for each pseudocode.**

Exercise

- Write a pseudocode to find the factorial of a given number. 0! is always 1. Factorial of a negative number is not possible.

- Write a pseudocode to accept a decimal number. Display it in the binary form.

- Write a pseudocode to accept a binary number and display it in the decimal form.

- **Write a DRY RUN for each pseudocode.**

Exercise

- Write as many pseudocodes to generate the following series. In all the following cases, accept N:

  - 4, 16, 36, 64, … N
  - 1, -2, 3, -4, 5, -6, … N
  - 1, 4, 27, 256, 3125, … N
  - 1, 4, 7, 12, 23, 42, 77, … N
  - 1, 4, 9, 25, 36, 49, 81, 100, … N
  - 1, 5, 13, 29, 49, 77, … N

- Write a psuedocode to find the sum of all the prime numbers in the range n to m. Display each prime number and also the final sum.
- **Write a DRY RUN for each psuedocode.**

Exercise

- Write a pseudocode to do the following:
- Accept the item code, description, qty and price of an item. Compute the total for the item.
- Accept the user’s choice. If the choice is ‘y’ then accept the next set of inputs for a new item and compute the total. In this manner, compute the grand total for all the items purchased by the customer.
- If the grand total is more than Rs. 10,000/- then, the customer is allowed a discount of 10%.
- If the grand total is less than Rs. 1,000/- and the customer chooses to pay by card, then a surcharge of 2.5% is levied on the grand total.
- Display the grand total for the customer.

The For Loop

- General Format

```
For control-variable  initial-value to final-value
[step +/- value]
statement1;
statement2;
:
End For;
```

- Example

```
For i  1 to 10
display i;
End For;
```

- The default step value is 1 (an increment of 1)

Requirements & Properties of the For loop

- **Requirements**
    - A proper initial value to the control variable
    - A proper exit value
    - A proper increment / decrement to the control variable
    - Proper begin and end of the loop
- **Properties**
    - Pre-tested loop
        - Need not be executed once
    - Enters when evaluation condition is ‘true’, exits when ‘false’
    - Initialization, condition and inc / dec are handled by a single
    - statement
        - If the loop runs ‘n’ times
        - The initialization is done only once
        - The condition is tested (n+1) times
        - The inc / dec is done n times

Exercise

- Write as many pseudocodes to generate the following series. In all the following cases, accept N:

- 1, -2, 6, -15, 31, -56, … N
- 1, 1, 2, 3, 5, 8, 13, … N
- 1, -2, 4, -6, 7,-10, 10,-14… N
- 1, 5, 8, 14, 27, 49, … N

- Write a pseudocode to find Xn (x to the power of n). Accept X and n. Display the result.

- Write a pseudocode to display the reverse of a string.

- Write a pseudocode to check if the string is a palindrome

Nested For Loop

- General Format

```
For outer-variable ← initial-value to final-value [step +/- value]
    statement1;
    statement2;
     For inner-variable ← initial-value to final-value [step +/- value]
            statement1;
            statement2;
            :
    End For;
    :
End For;
```

- Example

```
For i ← 1 to 10
        For j ← 1 to 10
                display (i+j);
        End For;
End For;
```

- The inner loop variable is initialized for each execution of the outer loop

Exercise

- Write the pseudocodes to generate the following outputs. In all the following cases, accept N:

```
* * * * *
* * * * *
* * * * *
* * * * *
:
N rows
```

```
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
:
N rows
```

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
:
N rows
```

```
*
*   *
*   *   *
*   *   *   *
:
N rows
```

Exercise

- Write the pseudocodes to generate the following outputs. In all the following cases, accept N:

```
1
1 2
1 2 3
1 2 3 4
:
N rows
```

```
1
2 2
3 3 3
4 4 4 4
:
N rows
```

```
1
2 3
4 5 6
7 8 9 10
:
N rows
```

```
1
1 2
3 5 8
:
:
N rows
```

Exercise

- Write the pseudocodes to generate the following outputs. In all the following cases, accept N:

```
1
-4   9
-16 25 -36
:
:
N rows
```

```
1
1   2
6  24 120
:
:
N rows
```

```
      *
    *  *
  *  *  *
*  *  *  *
:
N rows
```

```
      *
    *  *  *
  *  *  *  *  *
*  *  *  *  *  *  *
:
N rows
```

Using arrays

- An array is a collection of elements of the same data type
  - It is known as a HOMOGENEOUS data structure
- General Format
  - **DECLARE ARRAY** array-name **OF** size **data-type**;
- Example
  - **Declare Array** a1 of 10 **integer**;
  - **Declare Array** str of N **char**;
- Example to store and retrieve 10 integers
  - Declare Array a1 of 10 integer;

```
for i ← 1 to 10
    accept a1[i];
end for;

for i ← 1 to 10
    display a1[i];
end for;
```

Exercise

- Write a pseudocode to store N elements in an array of integer. Display the elements. Accept a number to be searched. Display whether the number is found or not in the array (LINEAR SEARCH).

- Write a pseudocode to store N elements in an array of integer. Display the elements. Sort the elements. Accept a number to be searched. Display whether the number is found or not in the array using BINARY SEARCH.

Using two-dimensional arrays

- A 2D array is essentially an array of arrays.
- General Format
  - **DECLARE ARRAY** array-name **OF [**size,size**] data-type**;
- Example
  - **Declare Array** a1 of [10,10] **integer**;
  - **Declare Array** str of [M,N] **char**;
- Example to store and retrieve M * N integers from a matrix

```
M ← 3 (integer);
N ← 3 (integer);
Declare Array a1 of [M,N] integer;
for i ← 1 to M
   for j ← 1 to N
         accept a1[i, j];
   end for;
end for;

for i ← 1 to M
   for j ← 1 to N
         display a1[i, j];
   end for;
   display;
```

Exercise

- Write a pseudocode to store elements into a M * N matrix of integer. Display the matrix and its transpose.
- Write a pseudocode to store elements into a N * N matrix of integer. Display whether it is an identity matrix or not.
- Write a pseudocode to store elements into a N * N matrix of integer. Display whether it is a symmetric matrix or not.

Question Time

?