RIZAL TECHNOLOGICAL UNIVERSITY
COLLEGE OF ENGINEERING ARCHITECTURE AND TECHNOLOGY

DATA STRUCTURE AND ALGORITHM

Algorithm Efficiency

There are often many approaches (algorithms) to solve a problem.
How do we choose them.

At the heart of computer program design are two (sometimes conflicting) goals:

- To design an algorithm that is easy to understand, code and debug
- To design an algorithm that makes efficient use of computer's resources.

Goal (1) is the concern of Software Engineering
Goal (2) is the concern of Data Structures and algorithm analysis.

How to Measure Efficiency?

- Empirical comparison (run programs).
- Asymptotic Algorithm Analysis (Big-Oh, Theta Notation).

Critical Resources:

- Time
- Space (disk, RAM)
- Programmer's Effort
- Ease of use (user's effort)

Factors affecting running time:

- Machine Load
- Operating System (OS)
- Compiler
- Problem size or Specific input values given problem size

***For most algorithms, running time depends on "size" of the input.

Running time is expressed as T(n) for some function T on input size n.

ANALYSIS of ALGORITHMS

- determining the amount of resources necessary to execute it such as time and storage
- usually in terms of CPU time and memory requirements

Analysis of Algorithms:

- Best-case analysis - can be used to improve accuracy of an overall worst-case analysis.
- Worst-case analysis - that is often of particular concern in real-time computing since it is crucial to know how much time will be needed in the worst case in order to guarantee that your algorithm will finish in time.
- Average-case analysis - It is difficult to determine average input means and it has properties that makes it difficult to mathematically characterize (for instance, consider, algorithms that are designed to operate on strings of text).

***analyzing algorithms is usually in terms of CPU time and memory requirements.

PSEUDOCODE

- Textual representation of flowchart
  - close to a natural language
  - the control structures impose the logic
- May become part of the program documentation
- Could be translated into a program

STEPWISE REFINEMENT

- The process by which a programmer refines an initial idea to a problem's solution into more specific terms.
- The last phase of refinement results in a program ready to be coded for execution.

RULES in PSEUDOCODE

The rules of Pseudocode are reasonably **straightforward**. All statements showing "dependency" are to be indented. These include while, do, for, if, switch.

For looping and selection, The keywords that are to be used include:
Do While...EndDo;
Do Until...Enddo;
While .... Endwhile is acceptable.

Also, Loop .... endloop is also VERY good and is language independent. Case...EndCase; If...Endif; Call ... with (parameters); Call; Return ....; Return; When;

As verbs, use the words **Generate, Compute, Process**, etc. Words such as **set, reset, increment, compute, calculate, add, sum, multiply, ... print, display, input, output, edit, test , etc.** with careful indentation tend to foster desirable pseudocode. Also, using words such as **Set and Initialize,** when assigning values to variables is also desirable.

RULES in PSEUDOCODE

- Always use scope terminators for loops and iteration.
- **INDENTATION in pseudocode should be identical to its implementation in a programming language.**
- As noted above, the pseudocode entries are to be cryptic, AND SHOULD NOT BE PROSE. NO SENTENCES.
- Do not include data declarations in your pseudocode.
- But do cite variables that are initialized as part of their declarations. E.g. "initialize count to zero" is a good entry

Examples

```
If student's grade is greater than or equal to 60
        Print "passed"
else
        Print "failed"
endif
```

```
Set total to zero
Set grade counter to one
While grade counter is less than or equal to ten
        Input the next grade
        Add the grade into the total
endwhile
Set the class average to the total divided by ten
Print the class average
```

RIZAL TECHNOLOGICAL UNIVERSITY
COLLEGE OF ENGINEERING ARCHITECTURE AND TECHNOLOGY

Thank You 😊
Keep safe
and God bless!