SWE1301: Introduction to Problem Solving and Software Development

Lecture 07 : Pseudocode & Algorithms
At: CIT Theatre
12-1pm
By: M .I. Mukhtar

Lecture Outline

- Pseudocode
- Algorithm

Pseudocode

- Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm.
- Pseudocode is similar to the algorithm without the numbers and somewhat condensed.
  - algorithm will be treated in the next few slides

Example 3: Pseudocode

From Example 1;
- Enter the radius and PI
- Calculate Area
- Display Area

From Example 2;
- Enter hours worked and Rate
- Calculate Pay
- Display Pay

Algorithms

- After using the PAC and the Pseudocode, the next step in organizing a solution is for the programmer to develop sets of **instructions** for the computer, called **algorithms**
- An Instruction is an order given to a computer processor by a computer program.
  - Computer programs are written by a computer programmer.
- The differences between an instruction in one computer language or application and another is in the **Syntax.**
  - Syntax refers to the rules governing the computer operating system, the language, and the application.

Algorithms

- Writing incorrect instructions to the computer will lead to either of the following:
  - an error message,
  - wrong answer,
  - no answer at all.
- Although a set of instructions must be in a correct order to lead to the correct result, there may be several "correct" order.

Algorithms

- Algorithm is defined as an unambiguous and precise set of steps for solving a problem (or sub-problem) in a predetermined amount of time using a finite amount of data.
  - Algorithm refers to the sequence of instructions that must be followed to solve a problem.
- Algorithms are generally created independent of underlying languages:
  - an algorithm can be implemented in more than one programming language.
  - For each problem or class of problems, there may be many different algorithms.

7

Algorithms…

- Setting up the algorithms is probably the hardest part of problem solving on the computer because;
  - The instructions **cannot assume anything**, **cannot skip steps**, must be in the **correct order,** must be executable **one step at a time**, and **must be complete**.
    - If the instructions are not properly ordered, the computer will, nevertheless, execute them in the order given, and the result will be wrong.

8

Properties of Algorithms…

- **Input**: A number of quantities are provided to an algorithm initially before the algorithm begins.
- **Definiteness**: Each step must be clear and unambiguous.
- **Effectiveness**: Each step must be carried out in finite time.
- **Output**: An algorithm must have output.
- **Correctness**: Correct set of output values must be produced from the each set of inputs.

9

Good Algorithms…

- A good algorithm should be:
  - Simple: An algorithm should not have no Unnecessary steps and no unnecessary complexity
  - Complete account for all inputs & cases
  - Precise: An algorithm should provide only one way to interpret the instruction.

10

Example 4: Algorithms…

From Example 1:

- Step 1: Enter radius
- Step 2: Enter PI
- Step 3: Area = PI * r * r
- Step 4: Print Area
- Step 5 : End

Or

- Step 1: Enter radius
- Step 2: Enter PI
- Step 3: Area = PI * pow(r,2)
- Step 4: Print Area
- Step 5 : End

11

Example 5: Algorithms…

From Example 2:

- Step 1: Enter Hours
- Step 2: Enter Rate
- Step 3: Gross_Pay = Hours * Rate
- Step 4: Print Gross_Pay
- Step 5 : End

12

Lecture Summary

- The efficiency of the computer relies on the effectiveness of the programmer or the user.
- Certain organizational tools such pseudocode and algorithm will help programmers to learn how to solve problems on the computer.

Questions !!!