CIS 1101 – PROGRAMMING 1

ALGORITHM: PSEUDOCODE

A **step-by-step procedure** for **solving a problem** especially in **mathematics or computing** in a **limited** number of steps.

ALGORITHM: IMPORTANT NOTES

❖ It has a definite beginning and a definite end.

❖ It has a finite number of steps.

❖ The instructions for each step are precise.

❖ Many algorithms involve repeating the same steps several times and can be carried out by a computer.

❖ Used for data processing, calculation and other related computer and mathematical operations.

❖ Used to manipulate data in various ways, such as inserting a new data item, searching for a particular item or sorting an item.

❖ The word "**algorithm**" comes from the name of a Persian mathematician called Al-Khwarizmi (Persian, 780-850).

1. **Pseudocode**

2. **Flowchart**

3. **Program**

REPRESENTATIONS OF ALGORITHM: PSEUDOCODE

- A **step-by-step instructions** written **in English-like statements**.

- An **artificial and informal language** that **helps to develop algorithms**.

REPRESENTATIONS OF ALGORITHM: PSEUDOCODE

❖ It is **useful** for **developing algorithms**
  ▪ that **will be converted to programs**

❖ A **convenient** and **user-friendly language**

  ▪ **not** an actual computer programming language

PSEUDOCODE: IMPORTANT NOTES

❑ Pseudocode programs are **not executed on computers**.

❑ They merely help one "**thinks out**" a program before attempting to write it in a programming language such as C.

❑ Pseudocode **consists only of action statements** (those that are executed when the program has been converted from pseudocode to C and is run in C).

3 TYPES OF STATEMENTS

1. **Composition/Sequential**
   - Sequence of statements are executed in order of appearance
   - **Example:** Finding Average of 5 Numbers Problem

2. **Alternation/Conditional**
   - Two or mores sequences of statements may form alternatives so that **only 1** of the alternatives is executed. Use *if-then* or *if-then-else* keywords.
   - **Example:** Determining Positive or Negative Number Problem

3. **Iteration**
   - A sequence of statements may be executed repeatedly, zero or more times (zero meaning execution may be omitted altogether). Use *while* keyword.
   - **Example:** Displaying numbers from 1 to 10

Purpose: For clarity and precision

BASIC OPERATIONS AND SYMBOLS

1. **Addition:** plus (+)
2. **Subtraction:** dash or hyphen (-)
3. **Multiplication:** asterisk (*)
4. **Division:** slash (/)
5. **Modulo:** per cent (%)
   - Remainder of a Division operation
   - Examples:
     - 5 % 2 = 1
     - 8 % 3 = 2
     - 5 % 8 = 5

ASSIGNMENT AND RELATIONAL OPERATIONS

1. **Assignment:** equal symbol (=)

2. **Relational Operators:**
   - **used to compare numbers and is either 1 (TRUE) or 0 (FALSE)**

   a) Less than: **<**

   b) Greater than: **>**

   c) Equality: **= =**

   d) Not equal: **!=**

   e) Less than or equal to: **<=**

   f) Greater than or equal to: **>=**

Note: See the difference between = and ==

LOGICAL OPERATORS

Logical Operator	Name	Description
&&	logical AND	It returns true when both conditions are true
||	logical OR	It returns true when at-least one of the condition is true
!	logical NOT	It is used to reverse state of its operand. If a condition is true, then Logical NOT operator will make it false.

LOGICAL OPERATORS

TRUTH TABLE

A	B	(A and B)	(A or B)	not(A and B)	not(A or B)
True	True	True	True	False	False
True	False	False	True	True	False
False	True	False	True	True	False
False	False	False	False	True	True

>>> (8>9) && (2<9)

    False

>>> (2>1) && (2>9)

    False

>>> (2==2) || (9<20)

    True

>>> (3!=3) || (9>20)

    False

>>> ! (8 > 2)

    False

>>> ! (2 > 10)

    True

1) Integer Division:

- **Integers** are numbers with no fractional component

- **Operands** and **quotient** are integers

- Examples:
  - 5 / 2 = 2
  - 6 / 2 = 3
  - 2 / 5 = 0

2) Floating-point Division:

- **Float** numbers have fractional numbers

- **Operands** and **quotient** are floating numbers

- examples:
  - $5.0 / 2.0 = 2.5$
  - $6.0 / 2.0 = 3.0$
  - $2.0 / 5.0 = 0.4$

WHAT IS A VARIABLE?

- A **storage location**
- paired with an **associated symbolic name**,
- which **contains some known or unknown quantity of information**
- referred to as a **value**.

☐ **Holder of values**