# COMPUTER SCIENCE

11

## ASSIGNMENT – 3

## ALGORITHMIC PROBLEM SOLVING

J. KAVITHA, B.Sc, B.Ed, M.C.A, M.Phil.,
Computer Instructor Gr - I
GHSS, S.S.KULAM
Coimbatore – 641107.

# ALGORITHMIC PROBLEM SOLVING

## Algorithms :

- An algorithm is a sequence of instructions to accomplish a task or solve a problem. An instruction describes an action. When the instructions are executed, a process evolves, which accomplishes the intended task or solves the given problem.

## Algorithmic Problems:

- There are some principles and techniques for constructing algorithms. We usually say that a problem is algorithmic in nature when its solution involves the construction of an algorithm. Some types of problems can be immediately recognized as algorithmic.

  **Example: The well-known Goat, grass and wolf problem**
- However, some algorithmic problems do not require us to construct algorithms. Instead, an algorithm is provided and we are required to prove some of its properties.

  **Example: The Chameleons of Chromeland problem**

## Building Blocks of Algorithms:

- We construct algorithms using basic building blocks such as:

## Data:

- Algorithms take input data, process the data and produce output data. Computers provide instructions to perform operations on data. For example, there are instructions for doing arithmetic operations on numbers, such as add, subtract, multiply and divide. There are different kinds of data such as numbers and text.

## Variables:

- Variables are named boxes for storing data. When we do operations on data, we need to store the results in variables. The data stored in a variable is also known as the value of the variable. We can store a value in a variable or change the value of variable, using an assignment statement.

## Control flow:

- An algorithm is a sequence of statements. However, after executing a statement, the next statement to be executed need not be the next statement in the algorithm. The statement to be executed next may depend on the state of the process. Thus, the order in which the statements are executed may differ from the order in which they are written in the algorithm. This order of execution of statements is known as the control flow.
- There are three important control flow statements to alter the control flow depending on the state.
  - Sequential control flow
  - Alternative control flow
  - Iterative control flow

## Functions

- Algorithms can become very complex. The variables of an algorithm and dependencies among the variables may be too many. Then, it is difficult to build algorithms correctly. In such situations, we break an algorithm into parts, construct each part separately and then integrate the parts to the complete algorithm.
- The parts of an algorithm are known as functions. A function is like a sub algorithm. It takes an input and produces an output, satisfying a desired input output relation.

## Algorithm Design Techniques:

- There are a few basic principles and techniques for designing algorithms.

## Specification:

- The first step in problem solving is to state the problem precisely. A problem is specified in terms of the input given and the output desired. The specification must also state the properties of the given input and the relation between the input and the output.

  **Example:**

  Let P be the required property of the inputs and Q the property of the desired outputs. Then the algorithm S is specified as
  - algorithm_name (inputs)
  - -- inputs : P
  - -- outputs: Q
- This specification means that if the algorithm starts with inputs satisfying P, then it will finish with the outputs satisfying Q. A double dash -- indicates that the rest of the line is a comment.

## Abstraction:

- A problem can involve a lot of details. Several of these details are unnecessary for solving the problem. Only a few details are essential. Ignoring or hiding unnecessary details and modeling an entity only by its essential properties is known as abstraction. For example, when we represent the state of a process, we select only the variables essential to the problem and ignore inessential details.

  **Example:**

- To ride a bicycle, it is sufficient to understand the functioning of the pedal, handlebar, brakes and bell. As a rider, we model a bicycle by these four features. A bicycle has a lot more details, which the rider can ignore. Those details are irrelevant for the purpose of riding a bicycle.

## Composition:

- An algorithm is composed of assignment and control flow statements. A control flow statement tests a condition of the state and depending on the value of the condition, decides the next statement to be executed.
- There are three important control flow statements:
  - **Sequential** * **Alternative** * **Iterative**

## Sequential statement:

- A sequential statement is composed of a sequence of statements. The statements in the sequence are executed one after another, in the same order as they are written in the algorithm and the control flow is said to be sequential.

## Alternative statement:

- A condition is a phrase that describes a test of the state. If C is a condition and both S1 and S2 are statements, then

  if C

  S1

  else

  S2

  is a statement, called an alternative statement, that describes the following action:

  1. Test whether C is true or false.
  2. If C is true, then do S1; otherwise do S2.

## Iterative statement:

- An iterative process executes the same action repeatedly, subject to a condition C. If C is a condition and S is a statement, then

  while C

  S

  is a statement, called an iterative statement, that describes the following action:

  1. Test whether C is true or false.
  2. If C is true, then do S and go back to step 1; otherwise do nothing.
- The iterative statement is commonly known as a loop.

## Decomposition:

- We divide the main algorithm into functions. We construct each function independently of the main algorithm and other functions. Finally, we construct the main algorithm using the functions. When we use the functions, it is enough to know the specification of the function. It is not necessary to know how the function is implemented.

## Refinement :

- After decomposing a problem into smaller subproblems, the next step is either to refine the subproblem or to abstract the subproblem.

1. Each subproblem can be expanded into more detailed steps. Each step can be further expanded to still finer steps and so on. This is known as refinement.
2. We can also abstract the subproblem. We specify each subproblem by its input property and the input-output relation. While solving the main problem, we only need to know the specification of the subproblems. We do not need to know how the subproblems are solved.

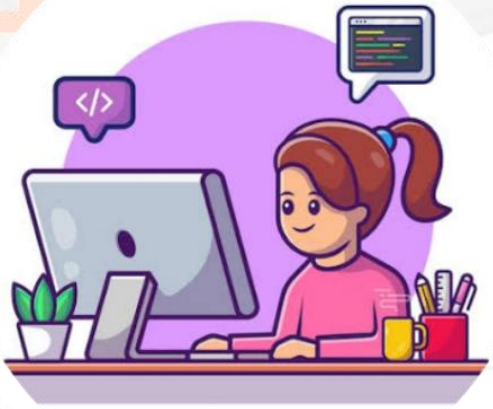
## Notations for Algorithms:

- We need a notation to represent algorithms. There are mainly three different notations for representing algorithms.
  - A programming language is a notation for expressing algorithms to be executed by computers.
  - Pseudo code is a notation similar to programming languages. Algorithms expressed in pseudo code are not intended to be executed by computers, but for communication among people.
  - Flowchart is a diagrammatic notation for representing algorithms. They give a visual intuition of the flow of control, when the algorithm is executed.


# EDUCATION IS THE

# KEY TO UNLOCK

# THE GOLDEN DOOR OF FREEDOM.



**J. Kavitha** B.SC.,B.Ed.,M.C.A.,M.Phil

**Computer Instructor Gr-1**

**GHSS, Sarkarsamakulam**

**Coimbatore - 641107.**