ALGORITHMS TO PROGRAMS:SPECIFICATION, TOP-DOWN DESIGN AND STEP-WISE REFINEMENT

BY: SHRESHTHA MISRA

[CSE DEPT]

Problem solving methodology

- The methodology to solve a problem is defined as the most efficient solution to the problem. Although, there can be multiple ways to crack a nut, but a methodology is one where the nut is cracked in the shortest time and with minimum effort.
- Under problem-solving methodology, we will see a step by step solution for a problem. These steps closely resemble the **software life cycle**. A software life cycle involves several stages in a program's life cycle.
- These steps can be used by any tyro programmer to solve a problem in the most efficient way ever.

Problem solving methodology

- The several steps of this cycle are as follows :
- **Step by step solution for a problem (Software Life Cycle):**

1. Problem Definition/Specification: A computer program is basically a machine language solution to a real-life problem. Because programs are generally made to solve the pragmatic problems of the outside world. In order to solve the problem, it is very necessary to define the problem to get its proper understanding. For example, suppose we are asked to write a code for " Compute the average of three numbers".

Problem solving methodology

- In this case, a proper definition of the problem will include questions like :
- "What exactly does average mean?"
  "How to calculate the average?"
- Once a problem has been defined, the program's specifications are then listed. Problem specifications describe what the program for the problem must do. It should definitely include :

INPUT :

- what is the input set of the program

OUTPUT :

- What is the desired output of the program and in what form the output is desired?

Problem solving methodology

- 2. Problem Analysis (Breaking down the solution into simple steps): This step of solving the problem follows a modular approach to crack the nut. The problem is divided into sub problems so that designing a solution to these sub problems gets easier. The solutions to all these individual parts are then merged to get the final solution of the original problem. It is like divide and merge approach.
- **Modular Approach for Programming :**
- The process of breaking a large problem into sub problems and then treating these individual parts as different functions is called modular programming.

Problem solving methodology

- Each function behaves independent of another and there is minimal inter-functional communication. There are two methods to implement modular programming :
- **Top Down Design :** In this method, the original problem is divided into subparts. These subparts are further divided. The chain continues till we get the very fundamental subpart of the problem which can't be further divided. Then we draw a solution for each of these fundamental parts.
- **Bottom Up Design :** In this style of programming, an application is written by using the pre-existing primitives of programming language.

Problem solving methodology

- These primitives are then amalgamated with more complicated features, till the application is written. This style is just the reverse of the top-down design style.
- **3. Problem Designing:** The design of a problem can be represented in either of the two forms :
- **Algorithm :** The set of instructions written down to formulate the solution of a problem is called algorithm. These instructions when followed to write the code produces the desired output. The steps involved in the algorithm are usually written in English or any colloquial language which is easier for the programmer to comprehend what the code needs.

Problem solving methodology

- **Flow charts:** Flow charts are diagrammatic representation of the algorithm. It uses some symbols to illustrate the starting and ending of a program along with the flow of instructions involved in the program.
- **Flowchart Symbols:**
- Different flowchart shapes have different conventional meanings. The meanings of some of the more common shapes are as follows:
- **Terminator:** The terminator symbol represents the starting or ending point of the system.

Problem solving methodology

- **Process**: A box indicates some particular operation.

- **Decision**: A diamond represents a decision or branching point. Lines coming out from the diamond indicates different possible situations, leading to different sub-processes.

Problem solving methodology

- **Data**: It represents information entering or leaving the system. An input might be an order from a customer. Output can be a product to be delivered.

- **On-Page Reference** : This symbol would contain a letter inside. It indicates that the flow continues on a matching symbol containing the same letter somewhere else on the same page.

Problem solving methodology

- **Off-Page Reference**: This symbol would contain a letter inside. It indicates that the flow continues on a matching symbol containing the same letter somewhere else on a different page.

- **Flow**: Lines represent the flow of the sequence and direction of a process.

Problem solving methodology

- **4. Coding: Once an algorithm is formed, it can't be executed on the computer. Thus in this step, this algorithm has to be translated into the syntax of a particular programming language. This process is often termed as 'coding'. Coding is one of the most important steps of the software life cycle. It is not only challenging to find a solution to a problem but to write optimized code for a solution is far more challenging.**
- **Writing code for optimizing execution time and memory storage :**
- The optimized code can save both time and memory.

Problem solving methodology

- **5. Program Testing and Debugging:** Program testing involves running each and every instruction of the code and check the validity of the output by a sample input. By testing a program one can also check if there's an error in the program. If an error is detected, then program debugging is done. It is a process to locate the instruction which is causing an error in the program and then rectifying it. There are different types of error in a program :
- **(i) Syntax Error:**
  Every programming language has its own set of rules and constructs which need to be followed to form a valid program in that particular language.

Problem solving methodology

- If at any place in the entire code, this set of rule is violated, it results in a syntax error.
- **Logical Error:**
- An error caused due to the implementation of a wrong logic in the program is called logical error. They are usually detected during the runtime.
- **Runtime Error:**
- Any error which causes the unusual termination of the program is called runtime error. They are detected at the run time.

Problem solving methodology

- **6. Documentation:** The program documentation involves the following:
- Problem Definition
- Problem Design
- Documentation of test perform
- History of program development
- User's manual A user's manual is used by a user to know about a program's input, processing, and output data.
- **7. Program Maintenance :** Once a program has been formed, to ensure its longevity, maintenance is a must. The maintenance of a program has its own costs associated with it, which may also exceed the development cost of the program in some cases

Problem solving methodology

- The maintenance of a program involves the following :
- Detection and Elimination of undetected errors in the existing program.
- Modification of current program to enhance its performance and
  adaptability.
- Enhancement of user interface
- Enriching the program with new capabilities.
- Updating of the documentation.

THANK YOU