Lecture 17

COMPUTER PROGRAMMING

Part 1

PROGRAM, PROGRAMMER & PROGRAMMING LANGUAGE

A computer program is a collection of instructions that performs a specific task when executed by a computer.

A computer program is usually written by a computer programmer in a programming language. Once it is written, the programmer uses a compiler to turn it into a language that the computer can understand and once the translation is done the computer can understand and execute these instructions provided in that program and achieve the program goal.

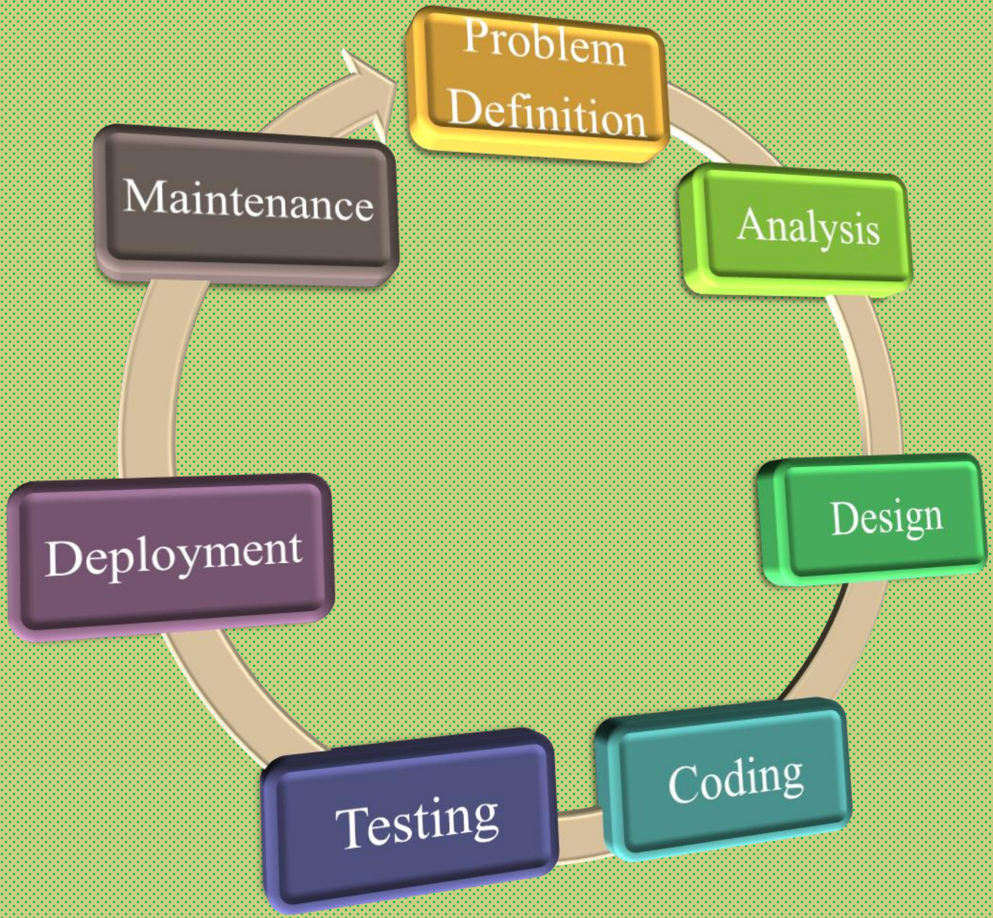
Softwares are collection of these programs and data files and have user interface to commute with the end user.

PROGRAM DEVELOPMENT STEPS

Development of program and software as system solution requires elaborate planning like any kind of project management. There are various models having 5 to 7 stages that describe software development in step by step method.

Program/Software/System Development Life Cycle (PDLC or SDLC) is such a systematic model that describes the development of quality software in 7 distinct phases.

PROGRAM DEVELOPMENT LIFE CYCLE (PDLC)



Seven Phases of PDLC/SDLC:

1. Problem Definition
2. Analysis
3. Design
4. Coding
5. Testing
6. Deployment
7. Maintenance

PROGRAM DEVELOPMENT LIFE CYCLE (PDLC)

1. **Problem Definition or Define the Problem:** This step involves the careful reading and re-reading of the problem until the programmer understands completely what is required.
2. **Analysis or Outlining the Solution:** Once the requirements are well understood, the developer team breaks the problem into simpler and smaller tasks and determine their solutions. This phase may require consultation with the customer regarding their need and the solutions prepared to fulfill them.

PROGRAM DEVELOPMENT LIFE CYCLE (PDLC)

3. **<u>Design or Develop the Outline into an Algorithm:</u>** Once the phase 2 is finalized, the developer team prepares algorithm that describes the tasks to be performed and the order in which they are to be carried out. Some important tasks of this phase are: **a)** Determining the data structure and data type to be used, **b)** Designing the user interface, **c)** Defining the inputs, **d)** Planning the output and **e)** Developing the algorithm to solve the tasks. Other program designing tools like Flowchart and Pseudo codes can also be used in this phase.

PROGRAM DEVELOPMENT LIFE CYCLE (PDLC)

4. **Coding the algorithm in specific programming language:** After completion of 3rd phase, the developer team starts writing the source code in required programming language. Before this a thorough planning regarding the language that will be used is necessary. The coding is done keeping in mind the future maintenance. This requires comments and labels inside the source code so that anyone capable of programming in that language can understand the code and do necessary modification. A supporting document describing the functionality of the code can be helpful.

PROGRAM DEVELOPMENT LIFE CYCLE (PDLC)

5. **Testing and Debugging for Errors:** After the completion of coding, the program is translated and tested with known inputs and the outputs are observed for possible errors and program bugs. A well deigned and carefully coded program requires less rounds of testing and debugging.
6. **Deployment or Deliver to Customer for Use:** The program is delivered to the customer with proper documents and user manuals. Sometimes in this phase a training for the end users is conducted. Customer feedbacks are used before going into full operation.

PROGRAM DEVELOPMENT LIFE CYCLE (PDLC)

7. **Maintenance and Upgrade:** This is the final phase that may redirect to the first phase if any addition, modification or program upgradation become a necessity. And then the newly found problems go through all the previous phases just like a cycle to solve the incompleteness. If the developed software proves to be flawless and can serve its needs without further upgradation, then the development cycle ends here.

PROGRAM DESIGN TOOLS

Necessity

In the 3rd phase of SDLC/PDLC, elaborately analyzed problem divided into various segments are coded by the team of developers. As it is difficult to start writing codes without any form of visual planning of the source code, it is often required to design the problem segments with the help from various program design tools.

Flowchart, Algorithm and Pseudo Codes are popular among program developers and these tools are used to build a solution structure of the program before the actual coding.

PROGRAM DESIGN TOOLS: ALGORITHM

An algorithm is an act for solving a problem in terms of the actions to be executed and the order in which those actions are to be executed. So this is basically is the sequence of some finite well defined simple steps taken to solve a problem.

Characteristics of an algorithm are:

- ❖ It must be finite number of steps taking finite time.
- ❖ Steps must be unambiguous.
- ❖ It can have zero or more inputs.
- ❖ It must have one or more outputs.
- ❖ It must be executable.

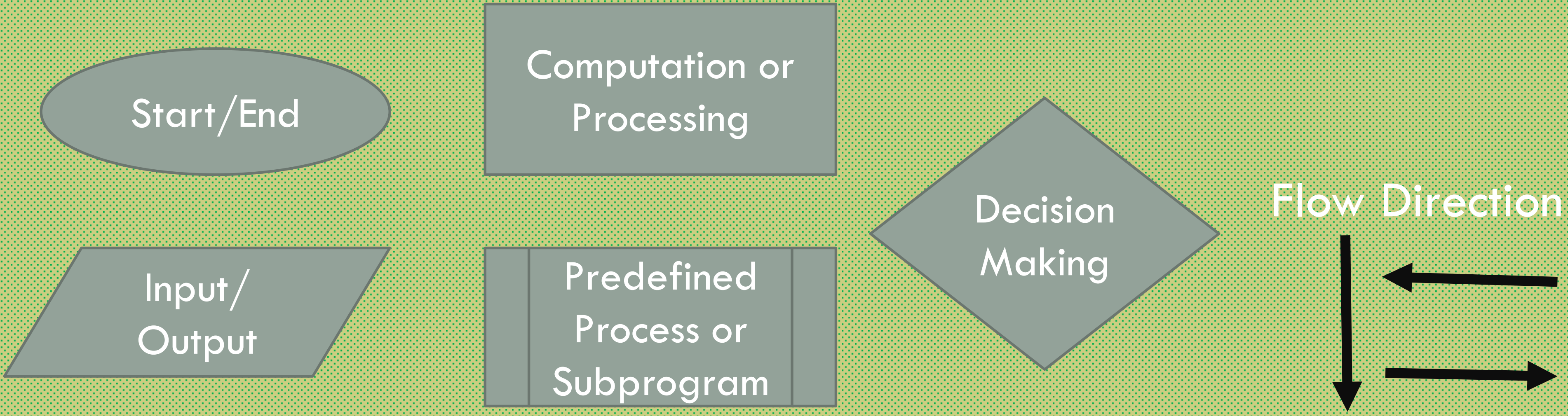
PROGRAM DESIGN TOOLS: ALGORITHM

Three Categories of Algorithm Operations

- **Sequential Operation** – Instruction steps are executed in sequential order, one by one, one after another.
- **Conditional Operation** – A control structure that asks true/false questions and then based upon the answer, the instruction which will be executed gets selected.
- **Iterative Operation (Loop)** – A control structure that repeats the execution of a block of instructions in a loop for a finite time based on program condition for reaching program goal.

PROGRAM DESIGN TOOLS: FLOWCHART

A flowchart or flow diagram is a graphical representation of a process or system that details the sequencing of steps required to create output. Flowchart is a picture of the separate steps of an algorithm with clear indication of step sequencing, branching and loops.



Start/End

Computation or Processing

Decision Making

Flow Direction

Input/ Output

Predefined Process or Subprogram

PROGRAM DESIGN TOOLS: FLOWCHART

Usefulness of Flowchart

- ✓ It is a better presentation of the designed problem for whole team.
- ✓ Most of the time, it is a better tool for effective analysis.
- ✓ Best option for program documentation.
- ✓ More friendly to the programmer during coding.
- ✓ Gives better guideline during debugging.
- ✓ During maintenance and upgradation, flowcharts give the best visual aid for the addition of code segments and code modification.

PROGRAM DESIGN TOOLS: FLOWCHART

Guidelines for Drawing Flowchart

- At first, list all the process in their logical order.
- It should be clear, easy to follow and without ambiguity.
- Top to bottom/left to right flow direction should be used.
- Only one flow line should go in a process box and only one should come out.
- In a decision making box, one line should go in and two/three lines should come out.
- Complex convergence is simplified using connectors.
- Must have a logical start and end point.
- It should be checked for validity with demo data.

PROGRAM DESIGN TOOLS: PSEUDO CODE

Pseudo code is a “grammar free” method of code writing which makes it very easy to write, read and understand. It is written in English (or coder’s native language) and looks like the actual source code in high level programming language.

The purpose of using pseudo code is that it is easier for even non programmer to understand the designed problem compared to the actual code. It is considered to be a form of designed program which has the appearance and transformability in between an independent algorithm and the actual source code in programming language.

PROGRAM DESIGN TOOLS: PSEUDO CODE

Guidelines for Writing Pseudo Code

- Statements are written in simple English.
- Each instruction is written on a separate line.
- Keywords and indentation are used to signify particular control structure.
- Each set of instruction is written from top to bottom with only one entry and one exit.
- Groups of statements may be formed into modules and can be labeled to identify the group feature.

PROGRAM DESIGN TOOLS: PSEUDO CODE

A pseudo code demo that calculates weekly wages in terms of work hour and pay rate. Employees with 40+ work hour get a $50 extra.

Pseudo Code:

```
1.  get hours_worked
2.  get pay_rate
3.  if hours_worked < 40 then
            3.1 gross_pay=pay_rate * hours_worked
     else
            3.2 gross_pay=pay_rate * hours_worked + 50
4.  display gross_pay
5.  Halt
          variables:hours_worked, pay_rate, gross_pay
```

Stay Home, Stay Safe

Always put on a mask when you are in public!