Algorithm & Flowchart & Pseudo code

Staff Incharge: S.Sasirekha

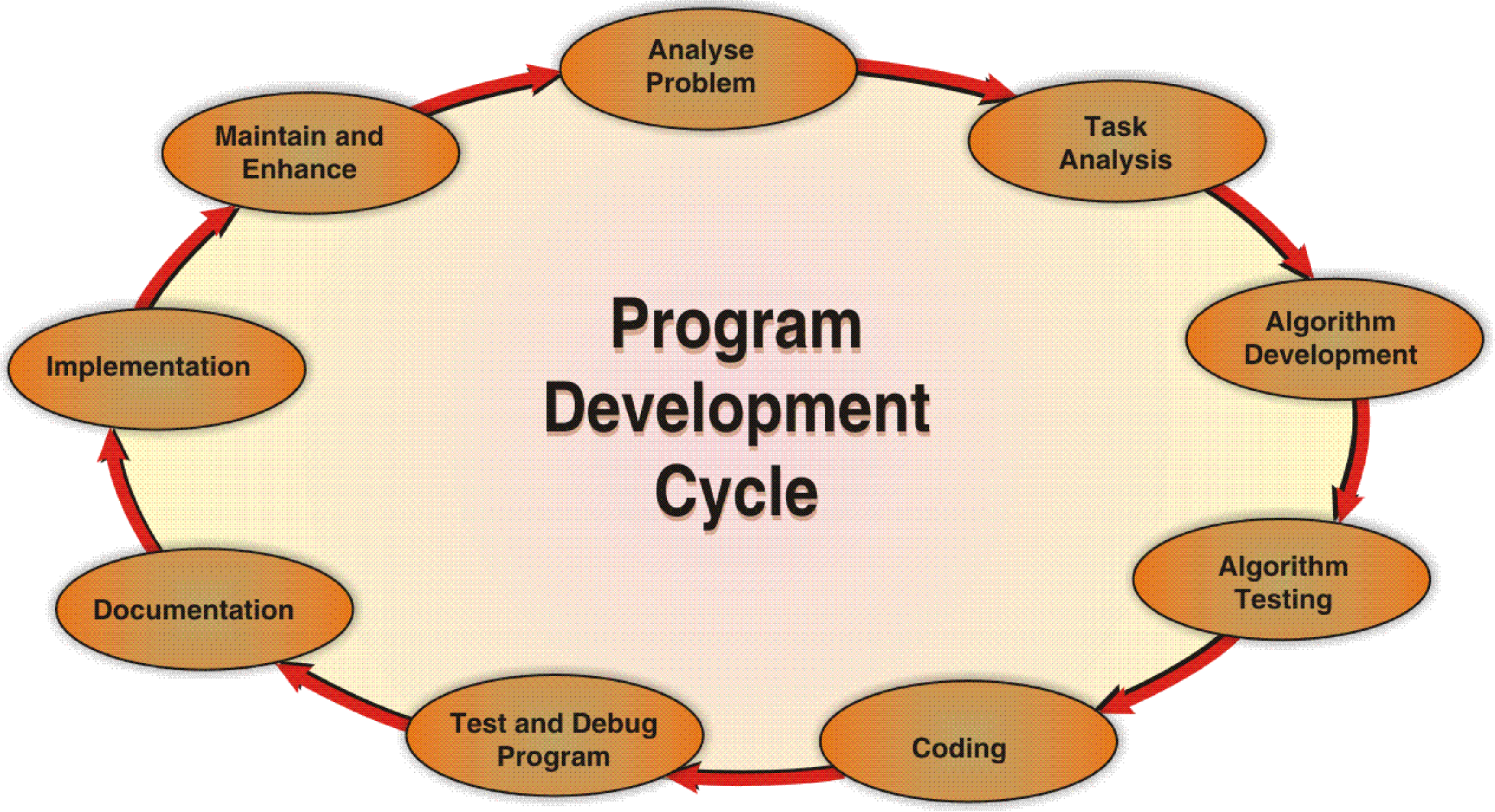
Computer Programming and Languages

- Computers work on a set of instructions called **computer program**, which clearly specify the ways to carry out a specific task.

Program Development Cycle

- Development cycle of a program includes:
  - **Analyse/Define the Problem**
  - **Task Analysis**
  - **Developing Algorithm**
  - **Testing the Algorithm for Accuracy**
  - **Coding the Solution**
  - **Test and Debug the Program**
  - **Documentation**
  - **Implementation**
  - **Maintenance and Enhancement**

Program Development Cycle



ALGORITHM

- An algorithm is defined as a finite sequence of explicit instructions, which when provided with a set of input values produces an output and then terminates.
- To be an algorithm, the steps must be unambiguous and after a finite number of steps, the solution of the problem is achieved.

FLOWCHART

- A flowchart is a pictorial representation of an algorithm in which the steps are drawn in the form of different shapes of boxes and the logical flow is indicated by interconnecting arrows.

- The boxes represent operations and the arrows represent the sequence in which the operations are implemented.

Benefits of Flowcharts

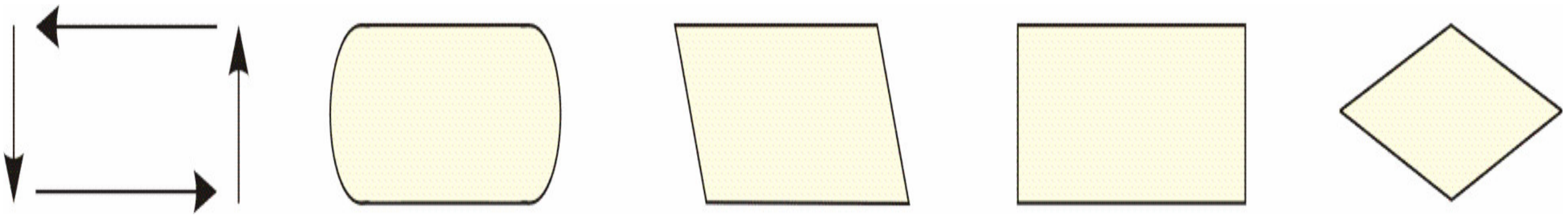
- Makes Logic Clear
- Communication
- Effective Analysis
- Useful in Coding
- Proper Testing and Debugging
- Appropriate Documentation

Limitations of Flowcharts

- Complex
- Costly
- Difficult to Modify
- No Update

Flowchart Symbols

- A flowchart uses special shapes to represent different types of actions or steps in a process.
- Some standard symbols, which are frequently required for flowcharts, are:



PSEUDOCODE

- Pseudo code is a generic way of describing an algorithm without using any specific programming language-related notations.

- It is an outline of a program, written in a form, which can easily be converted into real programming statements.

Benefits of Pseudocode

- Pseudocode provides a simple method of developing the program logic as it uses everyday language to prepare a brief set of instructions in the order in which they appear in the completed program.
- Language independent
- Easier to develop a program from pseudocode than a flowchart
- Easy to translate pseudocode into a programming language
- Pseudocode is compact and does not tend to run over many pages
- Pseudocode allow programmers who work in different computer languages

Limitations of Pseudo code

- The main disadvantage is that it does not provide visual representation of the program’s logic.
- No accepted standards style of writing pseudo codes.
- Cannot be compiled or executed and there are no real formatting or syntax rules.

PROGRAM CONTROL STRUCTURES

- Program statements that affect the order in which statements are executed, or that affect whether statements are executed, are called control structures. There are three control structures. They are sequential, selection, and repetition.
- **Sequence** construct is a linear progression where one task is performed sequentially after another.
- **Selection (Decision)** structure allows the program to make a choice between two alternate paths, whether it is true or false.
- **Repetition (Looping)** causes an interruption in the normal sequence of processing and directs the system to loop back to a previous statement in the program, repeating the same sequence again.

PROGRAMMING PARADIGMS

- Programming paradigm refers to how a program is written in order to solve a problem. Programming can be classified into three categories:

  1. **Unstructured Programming**
  2. **Structural Programming**
  3. **Object-Oriented Programming**

Unstructured Programming

- Unstructured style of programming refers to writing small and simple programs consisting of only one main program.
- All the actions such as inputs, outputs, and processing are done within one program only.

Structural Programming

- Using structural programming, a program is broken down into small independent tasks that are small enough to be understood easily, without having to understand the whole program at once.
- Each task has its own functionality and performs a specific part of the actual processing.
- These tasks are developed independently, and each task can carry out the specified task on its own, without the help of any other task.
- When these tasks are completed, they are combined together to solve the problem.

Object-Oriented Programming

- Object-oriented programming is a style of computer programming, which promotes building of independent pieces of code that interact with each other.
- It allows pieces of programming code to be reused and interchanged between programs.