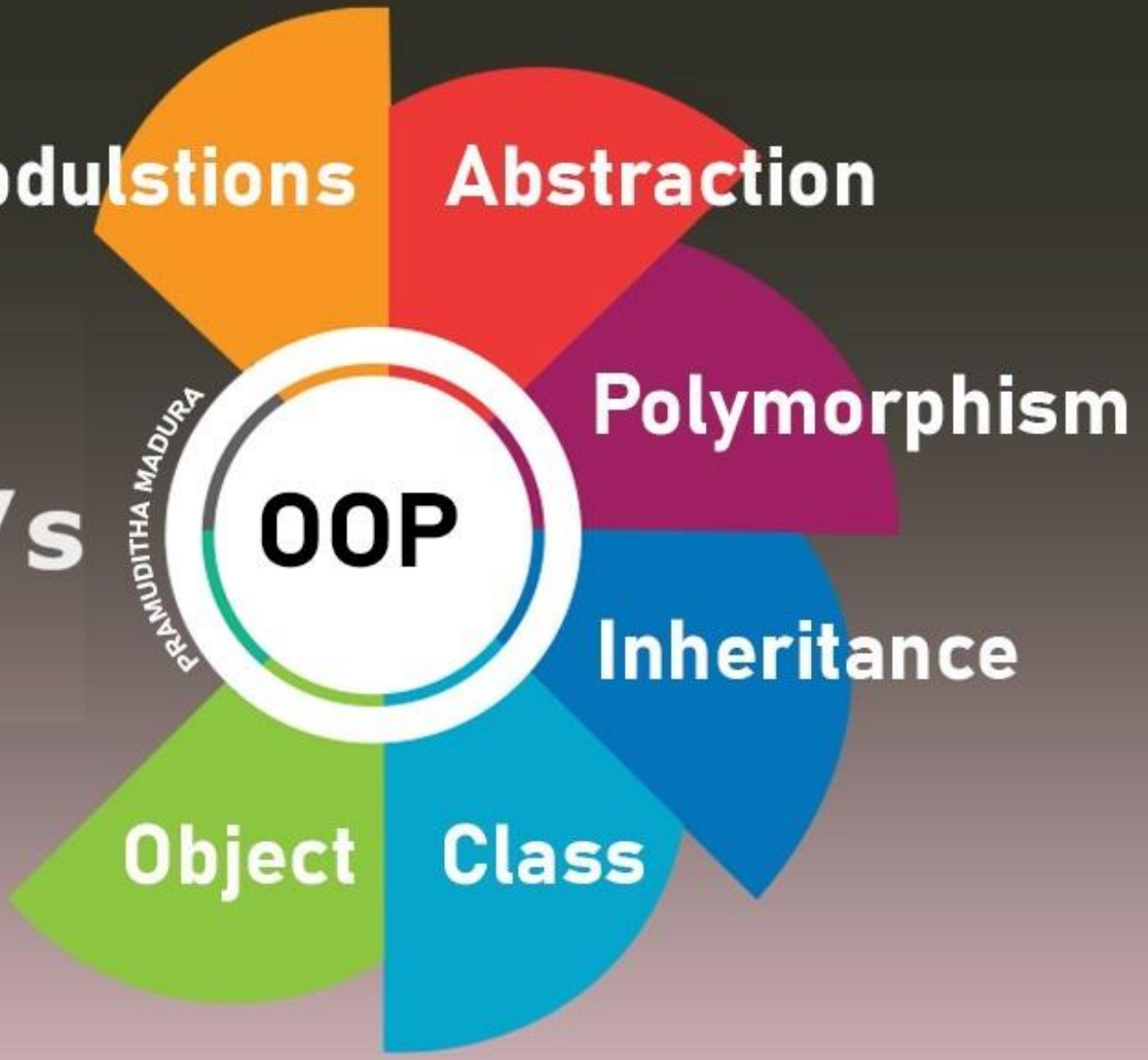
Procedural Programming v/s OOP



By – Mohammad Imran

Procedural Programming

Procedural programming is a programming paradigm built around the idea that programs are sequences of instructions to be executed. They focus heavily on splitting up programs into named sets of instructions called procedures, analogous to functions. A procedure can store local data that is not accessible from outside the procedure's scope and can also access and modify global data variables.

Some of the earliest procedural programming languages were **Fortran** and **ALGOL**. Ideas developed in ALGOL are very much relevant and prevalent in modern-day programming languages.

By – Mohammad Imran

Procedural Programming

Some tenets of procedural programming -

- ✓ Programs are composed of sequences of instructions. Minimal abstraction is present between the code and the machine.
- ✓ Procedures, which are logical blocks consisting of groups of instructions, can be invoked from other places in the code.
- ✓ A procedure can accept arguments and return values to the caller. Additionally, functions can access and modify variables in the global scope.
- ✓ Procedural languages follow structured programming practices and use block-based control flow rather than **goto** commands.

By – Mohammad Imran

Object Oriented Programming

Object-oriented Programming is a programming language that uses classes and objects to create models based on the real world environment. These objects contain data in the form of attributes and program codes in the form of methods or functions. In OOP, the computer programs are designed by using the concept of objects that can interact with the real world entities.

Examples of some object oriented programming languages are – **Java**, **C++**, **C#**, **Python**, **PHP**, **Swift**, etc.

By – Mohammad Imran

Procedural Versus Object Oriented Programming

Procedural Programming	Object Oriented Programming
This programming paradigm emphasizes on the use of functions where each function performs a specific task.	This programming paradigm is based on object oriented concept. Classes are used where instance of objects are created.
Fundamental elements used are variables and functions. The data in the functions are immutable (cannot be changed after creation).	Fundamental elements used are objects and methods and the data used here are mutable data.
Importance is not given to data but to functions.	Importance is given to data rather than procedures.

By – Mohammad Imran

Procedural Versus Object Oriented Programming

Procedural Programming	Object Oriented Programming
It follows declarative programming model.	It follows imperative programming model.
It uses recursion for iteration.	It uses loops for iteration.
It is parallel programming supported.	It does not support parallel programming.
The statements in this programming paradigm does not need to follow a particular order while execution.	The statements in this programming paradigm need to follow a order i.e., bottom up approach while execution.

By – Mohammad Imran

Procedural Versus Object Oriented Programming

Procedural Programming	Object Oriented Programming
Does not have any access specifier.	Has three access specifiers namely, Public, Private and Protected.
To add new data and functions is not so easy.	Provides an easy way to add new data and functions.
No data hiding is possible. Hence, Security is not possible.	Provides data hiding. Hense, secured programs are possible.

By – Mohammad Imran

Structure v/s Classes

- ✓ By default, all the members of the structure are public. In contrast, all members of the class are private.
- ✓ The structure will automatically initialize its members. In contrast, constructors and destructors are used to initialize the class members.
- ✓ When a structure is implemented, memory allocates on a stack. In contrast, memory is allocated on the heap in class.

By – Mohammad Imran

Structure V/s Classes

- ✓ Variables in a structure cannot be initialized during the declaration, but they can be done in a class.
- ✓ There can be no null values in any structure member. On the other hand, the class variables may have null values.
- ✓ A structure is a value type, while a class is a reference type.
- ✓ Operators to work on the new data form can be described using a special method.

By – Mohammad Imran

Problem Solving

By – Mohammad Imran

Problem Solving

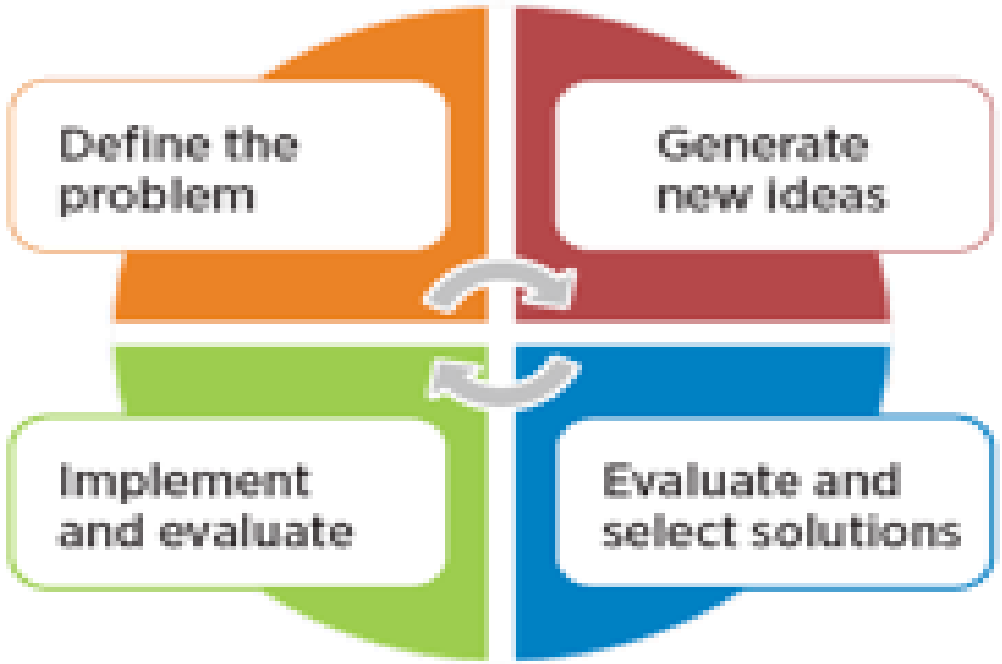
Introduction

Problem solving is a process of transforming the description of a problem into the solution of that problem by using our knowledge of the problem domain and by relying on our ability to select and use appropriate problem-solving Strategies, Techniques and Tools.

By – Mohammad Imran

Problem Solving

Problem solving is the act of defining a problem; determining the cause of the problem; identifying, prioritizing and selecting alternatives for a solution and implementing a solution.



By – Mohammad Imran

Problem Solving

Strategies

A problem-solving strategy is a plan used to find a solution or overcome a challenge. Each problem-solving strategy includes multiple steps to provide you with helpful guidelines on how to resolve a business problem or industry challenge. Effective problem-solving requires you to identify the problem, select the right process to approach it and follow a plan tailored to the specific issue you are trying to solve.

By – Mohammad Imran

Why is it important to understand multiple problem-solving strategies?

Understanding how a variety of problem-solving strategies work is important because different problems typically require you to approach them in different ways to find the best solution. By mastering several problem-solving strategies, you can more effectively select the right plan of action when faced with challenges in the future. This can help you solve problems faster and develop stronger critical thinking skills.

By – Mohammad Imran

Problem Definition

The computer is the symbol-manipulating machine that follows the set of instructions called a program.

A problem is a task to be performed. It is best thought of in terms of inputs and matching outputs. A problem definition should not include any constraints on how the problem is to be solved.

Any computing has to be performed independently without depending on the programming language and the computer.

By – Mohammad Imran

Problem Solving Strategies

Define Problem

The Problem-Definition Process encourages you to define and understand the problem that you're trying to solve, in detail. It also helps you confirm that solving the problem contributes towards your organization's objectives.

Taking the time to define a potential challenge can help you identify certain elements to create a plan to resolve them. Breaking down different areas and potential solutions to a problem can help you recognize how extensive the challenge could be and what strategies to put in place for a resolution.

By – Mohammad Imran

Problem Solving Strategies

Define Problem

For example, a company with a high employee turnover rate may focus on quickly hiring new employees to solve the immediate problem of being understaffed. If the hiring manager took the time to define the problem, they may realize that the ultimate reason they are understaffed is that their onboarding system makes it challenging for new hires to acclimate to the company culture. With this knowledge, the hiring manager may allocate additional resources to develop a more effective and welcoming onboarding process to increase employee retention.

By – Mohammad Imran

Problem Solving Strategies

Visualize the Problem

You might feel challenged when assessing the full scope of a problem or situation if you're closely involved with it. In these cases, try to visualize the problem by taking the time to focus on each individual element.

For example, if you're fixing a printer in your office that isn't working properly, you can visualize the different components of the printer, such as the paper tray or the ink cartridges, to determine the key issue. Once you identify this, your problem may be much easier to solve.

By – Mohammad Imran

Problem Solving Strategies

Draw Diagram for Problem

You might feel challenged when assessing the full scope of a problem or situation if you're closely involved with it. In these cases, try to visualize the problem by taking the time to focus on each individual element.

For example, if you're fixing a printer in your office that isn't working properly, you can visualize the different components of the printer, such as the paper tray or the ink cartridges, to determine the key issue. Once you identify this, your problem may be much easier to solve.

By – Mohammad Imran

Problem Solving Strategies

Break the Problem

It may be helpful to break larger problems down into smaller pieces or steps. This allows you to focus on resolving each smaller piece of the problem individually, which may be more manageable. Start by identifying what the requirements to solve this problem are.

You can ask yourself what you are trying to accomplish and what obstacles you need to overcome. Make a list of each relevant task you think of. Then organize each step by listing them in order of when they need to be accomplished. Finally, divide the list by assigning different tasks to individual members of your team.

By – Mohammad Imran

Problem Solving Strategies

Top Down and Bottom Up

The top **down** and **bottom** up are simple and common approaches. Yet the people do not know the real or the major differences among both of these approaches. Each of the approaches is not very difficult and very easy to understand.

Both the top-down and bottom-up approaches provide important advantages for the companies that control each approach. Both of the approaches are most commonly used for low-level and high-level work. But how each approach is used by the management is something that matters and might vary accordingly.

By – Mohammad Imran

Problem Solving Strategies

Top Down

In the terms of management, the top-down approach allows the head of the corporate to make the finalized decision. After the decision making it is then given a form of structural filter. The managers then tend to look out for knowledge and gather them in one single place. The gathered information is then analyzed and the conclusions are drawn.

The final decision is then forwarded to the rest of the team members which is then practiced by other employees. Many high-scale industries, for example, industries that manufacture goods integrate top-down method.

By – Mohammad Imran

Problem Solving Strategies

Top Down

How it works?

The top-down system works by making a decision through the knowledge held by the higher authorities and is then passed onto the other team members for to give their opinions on the deduced conclusion. The working of the top-down approach is very simple.

It is done by analyzing the decision and then the major decision is then taken by consulting the other team members as well. The top-down approach can be effective because it remains the same from project to project. The top-down approach is well-practiced and grows more efficient over time.

By – Mohammad Imran

No room for confusion

This management method (top-down) gives or predicts accurate results, due to which the method is more efficient. The reason for less room for confusion is due to the discussions that take in a single place with a good flow. Hence, there is very little chance of misunderstandings or confusion.

Good Management Style

This approach is easy to implement. The learning curve for the particular approach that is, a top-down approach is much wide and easier. If you are a team leader then you can easily adjust your other team members to the good management style of a top-down approach.

Helps in achieving goals more quickly

While practicing the top-down approach one major benefit is that it is not much time-consuming. The problems are sorted out more efficiently and are then looked into by the head for the conclusions that are drawn in much lesser time as compared to the bottom-up approach.

These are some of plus points of top down approach when talking about Top-Down vs. Bottom-Up: What's the Difference.

The team is not involved in the decision making

In this method, all team members concerning the management department cannot conclude decisions. It is the head of the company or the cooperate that takes the final decision and then passes it forward.

Risk of missing out on better and great ideas

The administration technique is based on the single decision by the head of the department and since the base of the top-down approach does not involve other employees like the bottom-up approach, it reduces major profitable opportunities.

Problem Solving Strategies

Bottom Up

The bottom-up approach first keeps its focus on solving the smaller problems and then integrating them into a whole and complete solution. The bottom-up approach is mostly practiced for many future tasks like goal setting, budgeting, and forecasting.

In this design, individual parts of the system are specified in detail. The parts are linked to form larger components, which are in turn linked until a complete system is formed. Object-oriented language such as **C++** or **Java** uses a bottom-up approach where each object is identified first.

By – Mohammad Imran

Problem Solving Strategies

Bottom Up | **Advantages**

Make decisions about reusable low-level utilities then decide how there will be put together to create high-level construct.

Problem Solving Strategies

Top Down and Bottom Up Differences

	TOP DOWN APPROACH	BOTTOM UP APPROACH
1.	In this approach We focus on breaking up the problem into smaller parts.	In bottom up approach, we solve smaller problems and integrate it as whole and complete the solution.
2.	Mainly used by structured programming language such as COBOL, Fortran, C, etc.	Mainly used by object oriented programming language such as C++, C#, Python.
3.	Each part is programmed separately therefore contain redundancy.	Redundancy is minimized by using data encapsulation and data hiding.
4.	In this the communications is less among modules.	In this module must have communication.
5.	It is used in debugging, module documentation, etc.	It is basically used in testing.
6.	In top down approach, decomposition takes place.	In bottom up approach composition takes place.

By – Mohammad Imran

Problem Solving Strategies

Top Down and Bottom Up Differences

SN.	TOP DOWN APPROACH	BOTTOM UP APPROACH
9.	**Pros-** •Easier isolation of interface errors •It benefits in the case error occurs towards the top of the program. •Defects in design get detected early and can be corrected as an early working module of the program is available.	**Pros-** •Easy to create test conditions •Test results are easy to observe •It is suited if defects occur at the bottom of the program.
10.	**Cons-** •Difficulty in observing the output of test case. •Stub writing is quite crucial as it leads to setting of output parameters. •When stubs are located far from the top level module, choosing test cases and designing stubs become more challenging.	**Cons-** •There is no representation of the working model once several modules have been constructed. •There is no existence of the program as an entity without the addition of the last module. •From a partially integrated system, test engineers cannot observe system-level functions. It can be possible only with the

Top Down and Bottom Up Differences

Top Down Pros –

- ✓ Easier isolation of interface errors
- ✓ It benefits in the case error occurs towards the top of the program.
- ✓ Defects in design get detected early and can be corrected as an early working module of the program is available.

Top Down Cons –

- ✓ Difficulty in observing the output of test case.
- ✓ Stub writing is quite crucial as it leads to setting of output parameters.
- ✓ When stubs are located far from the top level module, choosing test cases and designing stubs become more challenging.

By – Mohammad Imran

Top Down and Bottom Up Differences

Bottom Up Pros –

- ✓ Easy to create test conditions
- ✓ Test results are easy to observe
- ✓ It is suited if defects occur at the bottom of the program.

Bottom Up Cons –

- ✓ There is no representation of the working model once several modules have been constructed.
- ✓ There is no existence of the program as an entity without the addition of the last module.
- ✓ From a partially integrated system, test engineers cannot observe system-level functions. It can be possible only with the installation of the top-level test driver.

By – Mohammad Imran

Problem Solving Technique

The problem solving techniques involves the following steps:

- ✓ Define the problem
- ✓ Formulate the mathematical model
- ✓ Develop the algorithm
- ✓ Write the code for the problem
- ✓ Test the program

By – Mohammad Imran

Problem Solving Techniques

Define Problem

The Problem-Definition Process encourages you to define and understand the problem that you're trying to solve, in detail. It also helps you confirm that solving the problem contributes towards your organization's objectives.

Formulate Mathematical Problem

Any technical problem provided can be solved mathematically.

Full knowledge about the problem should be provided along with the underlying mathematical concept.

Example:

```
Data1 + Data2 / 2
(Data1 + Data2) / 2
```

Algorithm

"A step by step process is known as the algorithm."

An algorithm is any well-defined computational procedure that takes a value or set of values as input and produce an output. The algorithm is very important for programmers.

Problem Solving Techniques

Algorithm

To be a good programmer one must maintain the following basic sequence.

- ✓ Design of algorithm
- ✓ Prepare a flowchart for that algorithm
- ✓ Write a program according to flowchart
- ✓ Test the program by inputting different values.
- ✓ Implement the program.

By – Mohammad Imran

Algorithm Design

Algorithm for adding two numbers

Step – 1	-	**START**
Step – 2	-	Input A and B
Step – 3	-	C = 0
Step – 4	-	C = A + B
Step – 5	-	Print C
Step – 6	-	**END**

By – Mohammad Imran

Algorithm

Design

An algorithm to convert temperature from Centigrade to Fahrenheit

Step – 1	-	**START**
Step – 2	-	Input C
Step – 3	-	F = 0
Step – 4	-	F = 9 * C / 5 + 32
Step – 5	-	Print F
Step – 6	-	**END**

By – Mohammad Imran

Flowchart

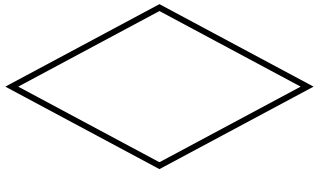
Flowchart is the pictorial representation of algorithm and the summary of the decisions and flows that makes up a procedure or process from beginning to end.

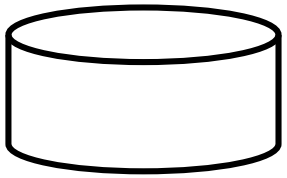
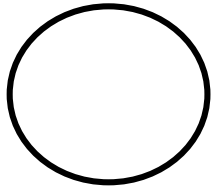
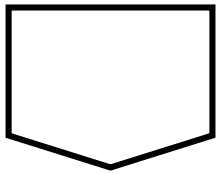
Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.

By – Mohammad Imran

Flowchart

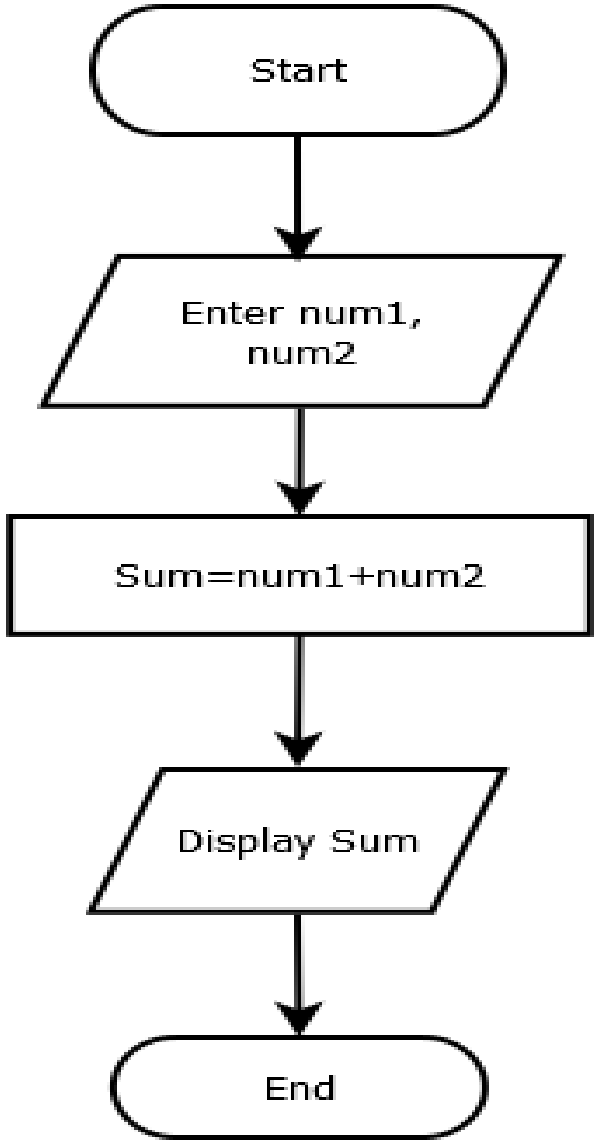
Symbols

Symbols	Meaning
	Terminal Start / End
	Decision Making
	Processing
	Input / Output

Symbols	Meaning
	Storage Disk
	Connector
	Different page connector
	Document / Hard Copy

By – Mohammad Imran

Flowchart Diagram



By – Mohammad Imran

Problem Solving Techniques

Solution

Write the code for the problem

- ✓ The algorithm created must be converted to any programming language.
- ✓ The compiler will convert the program code to the machine language which the computer can understand.

Test the program

- ✓ Testing involves checking errors both syntactically and semantically.
- ✓ The errors are called as “bugs”.
- ✓ When the compiler finds the bugs, it prevents compiling the code from programming language to machine language.
- ✓ Check the program by providing a set of data for testing.

Program

Definition

A program consists of a series of instructions that a computer processes to perform the required operation.

Set of computer programs that describe the program are called software.

The process of software development is called Programming and the person who creates the computer programs is called Programmer.

A program is a set of instruction written by the programmer. A collection of a programs are known as software.

By – Mohammad Imran

Program Requirements

Thus, in order to design a program, a programmer must determine three basic requirements:

- ✓ The instructions to be performed.
- ✓ The order in which those instructions are to be performed.
- ✓ The data required to perform those instructions.

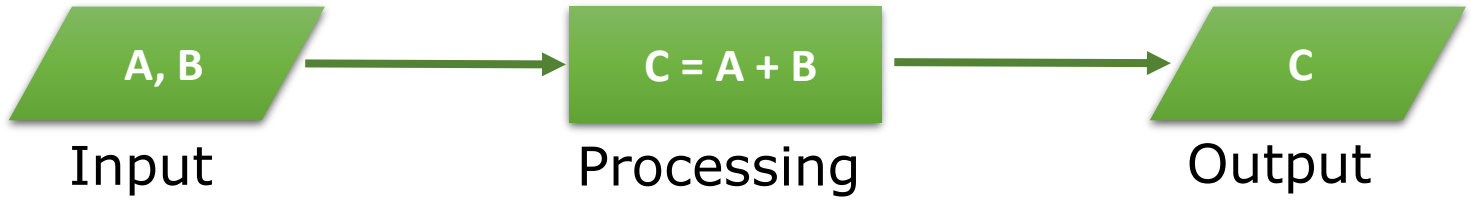
Program

Example - 1

Write a program to add two numbers

- ✓ Input two numbers.
- ✓ Add the two numbers.
- ✓ Display the output.

Suppose we want to calculate the sum of two numbers, **A** and **B**, and store the result in **C**. Here, A and B are the inputs, addition is the process and **C** is the output of the program.



By – Mohammad Imran

Program

Example - 1

```
#include<stdio.h>
int main()
{
      int A, B, C = 0;
      cout << "Enter A = ";
      cin >> A;
      cout << "Enter B = ";
      cin >> B;
      C = A + B;
      cout << "Sum C = " << C;
      return 0;
}
```

OUTPUT

```
Enter A = 20
Enter B = 10
Sum C = 30
```

By – Mohammad Imran

Program Characteristic

Any computer program should have the following characteristics.

- ✓ **Accuracy** – Any calculation made in the program should be correct and accurate.
- ✓ **Clarity** – It refers to the overall readability of the program which helps the user to understand the underlying program logic easily without much difficulty.
- ✓ **Modularity** – When developing any program, the task is divided into several modules or subtasks. These modules are developed independently i.e. each task does not depend on the outer task.

By – Mohammad Imran

Program Characteristic

- ✓ **Portability** – It is defined as the ability to run the application program on different platforms.
- ✓ **Flexibility** – Any program written should be flexible i.e. the program should be developed in such a way that it can handle most of the changes without rewriting the entire program.
- ✓ **Efficiency** – Any program needs certain memory and processing time to process the data. This memory and processing unit should be of least amount. This is the efficiency of the program.

Program Characteristic

- ✓ **Generality** – The program should be generic in nature. If a program is created for a specific task then it can be used for all the similar tasks in the same domain.
- ✓ **Documentation** – Any application program should be well-documented such that even in the absence of the developer and the author, the programmers can understand the concept behind it.

Program Development Life Cycle

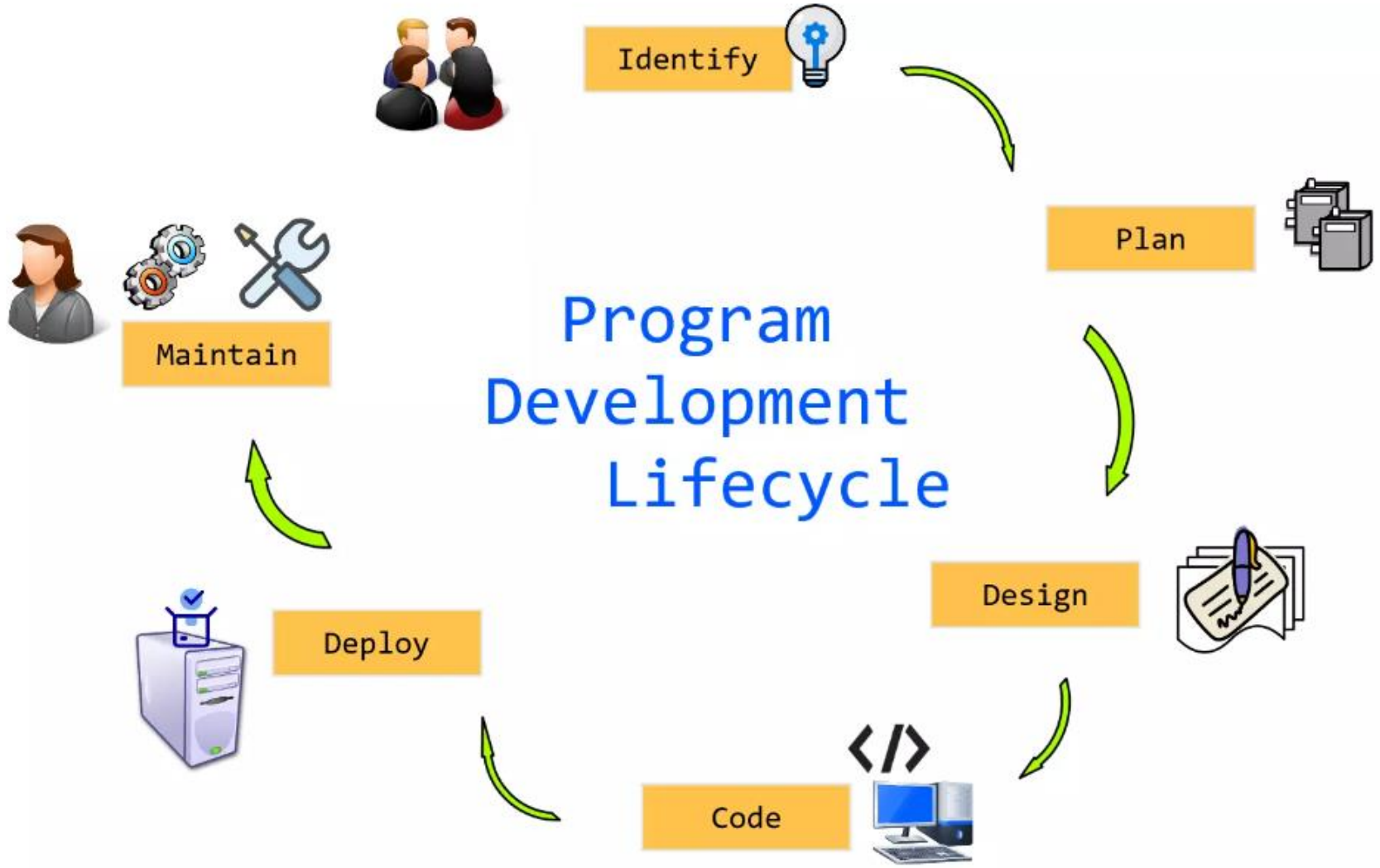
PDLC

The Program Development Life Cycle (**PDLC**) is a process used in software engineering to manage the development of software programs. The PDLC is similar to the Software Development Life Cycle **(SDLC)**, but is applied at a higher level, to manage the development of multiple software programs or projects.

By – Mohammad Imran

Program Development Life Cycle

PDLC



By – Mohammad Imran

Program Development Life Cycle

PDLC

The PDLC is typically divided into several phases, including:

- ✓ **Planning:** In this phase, the goals and objectives of the program are defined, and a plan is developed to achieve them. This includes identifying the resources required and determining the budget and schedule for the program.
- ✓ **Analysis:** In this phase, the requirements for the program are defined and analyzed. This includes identifying the stakeholders, their needs and expectations, and determining the functional and non-functional requirements for the program.

By – Mohammad Imran

Program Development Life Cycle

PDLC

- ✓ **Design:** In this phase, the program's architecture and design are developed. This includes creating a detailed design of the program's components and interfaces, as well as determining how the program will be tested and deployed.
- ✓ **Implementation:** In this phase, the program is developed and coded. This includes writing the program's source code and creating any necessary documentation.
- ✓ **Testing:** In this phase, the program is tested to ensure that it meets the requirements and is free of defects.

By – Mohammad Imran

Program Development Life Cycle

PDLC

- ✓ **Deployment:** In this phase, the program is deployed and made available to users.

- ✓ **Maintenance:** After the deployment, the program is maintained by fixing any bugs or errors that are found and updating the program to meet changing requirements.

Program Development Life Cycle

PDLC

Program Development Life Cycle (PDLC) is a systematic way of developing quality software. It provides an organized plan for breaking down the task of program development into manageable chunks, each of which must be successfully completed before moving on to the next phase.

By – Mohammad Imran

Program Development Life Cycle

PDLC

The program development process is divided into the steps discussed below:

- ✓ **Defining the Problem –** The first step is to define the problem. In major software projects, this is a job for system analyst, who provides the results of their work to programmers in the form of a program specification. The program specification defines the data used in program, the processing that should take place while finding a solution, the format of the output and the user interface.

By – Mohammad Imran

Program Development Life Cycle

PDLC

- **Designing the Program –** Program design starts by focusing on the main goal that the program is trying to achieve and then breaking the program into manageable components, each of which contributes to this goal. This approach of program design is called *top-bottom program design* or *modular programming*.

Program Development Life Cycle

PDLC

- ✓ **Coding the Program –** Coding the program means translating an algorithm into specific programming language. The technique of programming using only well defined control structures is known as *Structured programming*. Programmer must follow the language rules, violation of any rule causes *error*. These errors must be eliminated before going to the next step.

Program Development Life Cycle

PDLC

- ✓ **Testing and Debugging the Program –** After removal of syntax errors, the program will execute. However, the output of the program may not be correct. This is because of logical error in the program. A logical error is a mistake that the programmer made while designing the solution to a problem. So the programmer must find and correct logical errors by carefully examining the program output using *Test data*. Syntax error and Logical error are collectively known as *Bugs*. The process of identifying errors and eliminating them is known as *Debugging*.

Program Development Life Cycle

PDLC

- ✓ **Documenting the Program –** After testing, the software project is almost complete. This phase ends by writing a manual that provides an overview of the program's functionality, tutorials for the beginner, in-depth explanations of major program features, reference documentation of all program commands and a thorough description of the error messages generated by the program.

Program Development Life Cycle

PDLC

- **Deploying and Maintaining the Program –** In the final phase, the program is deployed (installed) at the user's site. Here also, the program is kept under watch till the user gives a green signal to it. Even after the software is completed, it needs to be maintained and evaluated regularly. In software maintenance, the programming team fixes program errors and updates the software.