Presented By :Mostafa Saqly

OOP Day1 Introduction

What is the Software Development Life Cycle?

Software Development Life Cycle is the application of standard business practices to building software applications. It's typically divided into six to eight steps : Planning, Requirements, Design, Build, Document, Test, Deploy, Maintain.

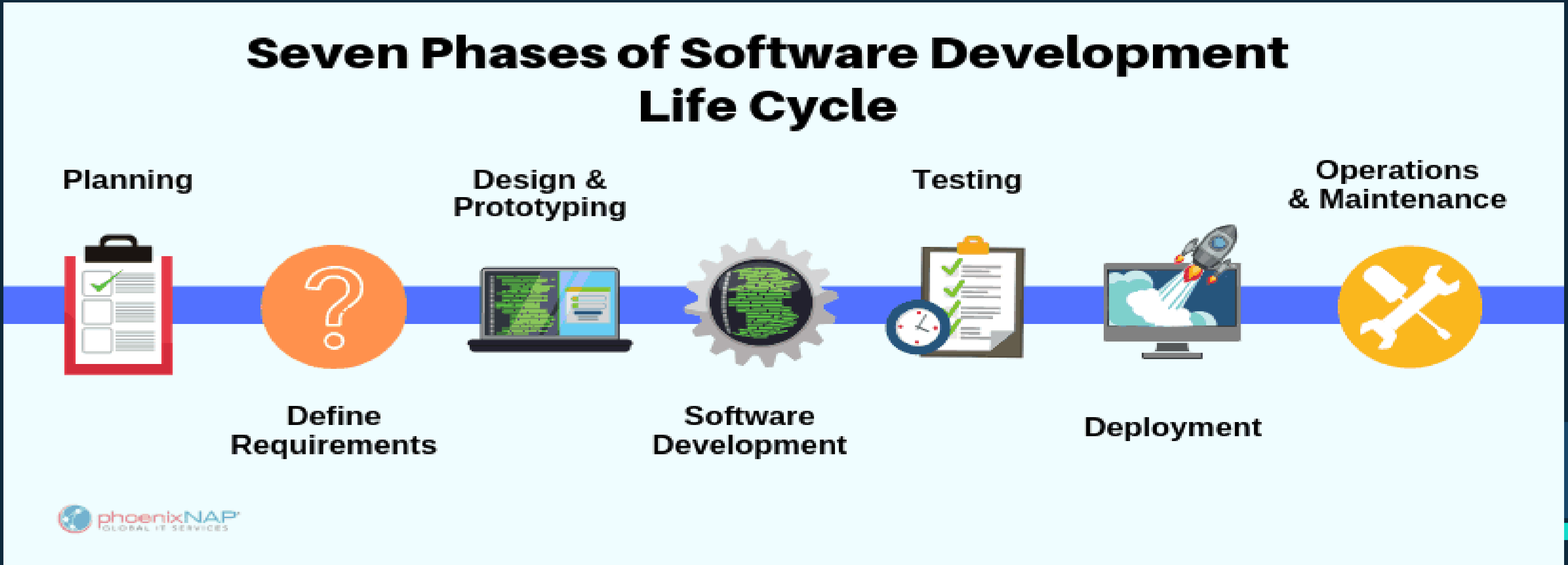
Some project managers will combine, split, or omit steps, depending on the project's scope. These are the core components recommended for all software development projects..

SDLC is a way to measure and improve the development process. It allows a fine-grain analysis of each step of the process. This, in turn, helps companies maximize efficiency at each stage. As computing power increases, it places a higher demand on software and developers. Companies must reduce costs, deliver software faster, and meet or exceed their customers' needs. SDLC helps achieve these goals by identifying inefficiencies and higher costs and fixing them to run smoothly.

How the Software Development Life Cycle Works?

The Software Development Life Cycle simply outlines each task required to put together a software application. This helps to reduce waste and increase the efficiency of the development process. Monitoring also ensures the project stays on track, and continues to be a feasible investment for the company.

The Seven Phases of the SDLC



Seven Phases of Software Development Life Cycle

Planning

Define Requirements

Design & Prototyping

Software Development

Testing

Deployment

Operations & Maintenance

1

Planning

n the Planning phase, project leaders evaluate the terms of the project. This includes calculating labor and material costs, creating a timetable with target goals, and creating the project’s teams and leadership structure.

Planning should clearly define the scope and purpose of the application

2

Define Requirements

Defining requirements is considered part of planning to determine what the application is supposed to do and its requirements.

Requirements also include defining the resources needed to build the project

3

Design and Prototyping

The Design phase models the way a software application will work some aspects of the design include : **Architecture , User Interface , Platforms , Programming , Communications , Security .** Prototyping can be a part of the Design phase. A prototype is like one of the early versions of software in the Iterative software development model. It demonstrates a basic idea of how the application looks and works.

4

Software development

This is the actual writing of the program. A small project might be written by a single developer, while a large project might be broken up and worked by several teams.

They also help ensure compatibility between different team projects and to make sure target goals are being met.

5

Testing

It’s critical to test an application before making it available to users.

Different parts of the application should also be tested to work seamlessly together—performance test, to reduce any hangs or lags in processing.

6

Deployment

In the deployment phase, the application is made available to users. Many companies prefer to automate the deployment phase. This can be as simple as a payment portal and download link on the company website. It could also be downloading an application on a smartphone.

7

Operations and Maintenance

At this point, the development cycle is almost finished. The application is done and being used in the field. The Operation and Maintenance phase is still important, though. In this phase, users discover bugs that weren’t found during testing. These errors need to be resolved, which can spawn new development cycles.

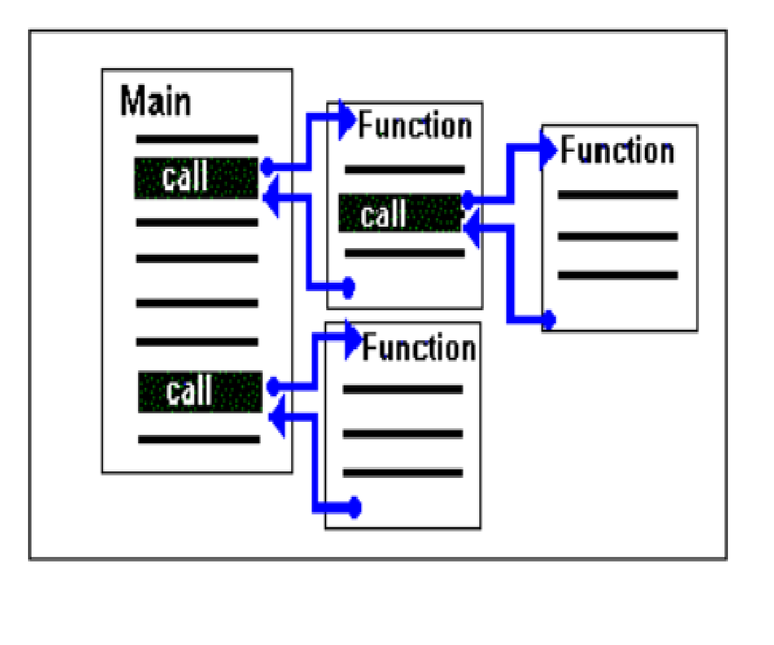
What IS Object Oriented Programming ?

- **Object-oriented programming (OOP) :** is a programming paradigm based on the concept of "objects“ .
- **A programming paradigm** : is a style of programming, a way of thinking about software construction.
- A programming paradigm does not refer to a specific language but rather to a way to build a program or a methodology to apply.
- Some languages make it easy to write in some paradigms but not others.
- Some Programming Languages allow the programmer to apply more than one Paradigm.

What IS Procedural Programming?

- Procedural programming (PP), also known as inline programming takes a top-down approach. It is about writing a list of instructions to tell the computer what to do step by step. It relies on procedures or routines.

Example of Procedural Programming?



Main

call

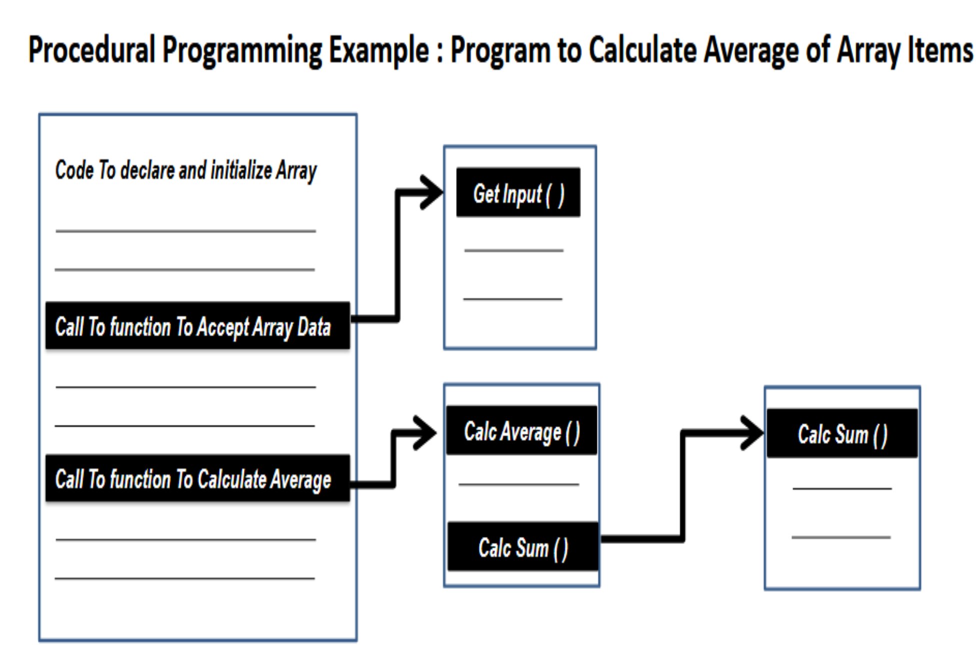
call

Function

call

Function

Function



Procedural Programming Example : Program to Calculate Average of Array Items

Code To declare and initialize Array

Call To function To Accept Array Data

Call To function To Calculate Average

Get Input ( )

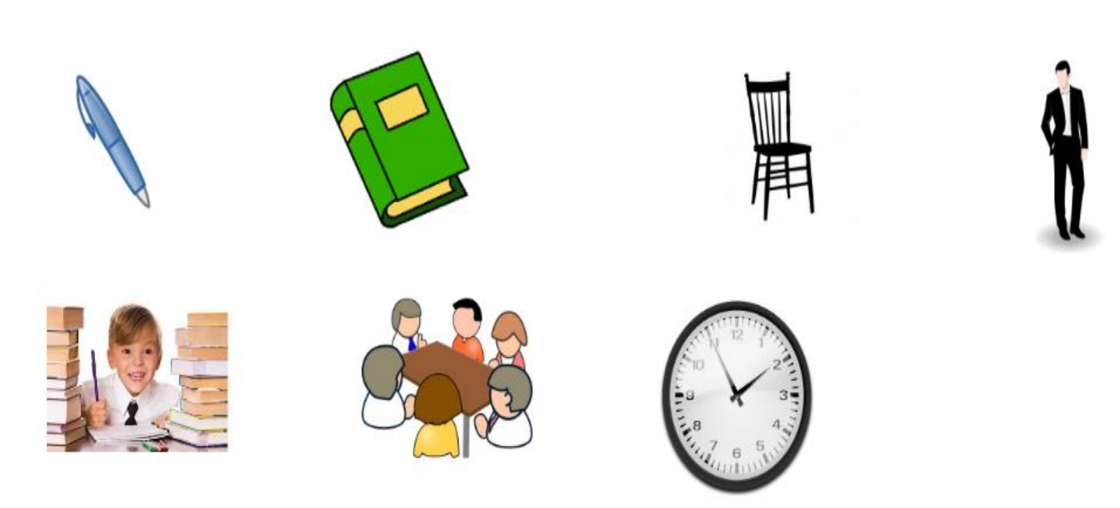
Calc Average ( )

Calc Sum ( )

Calc Sum ( )

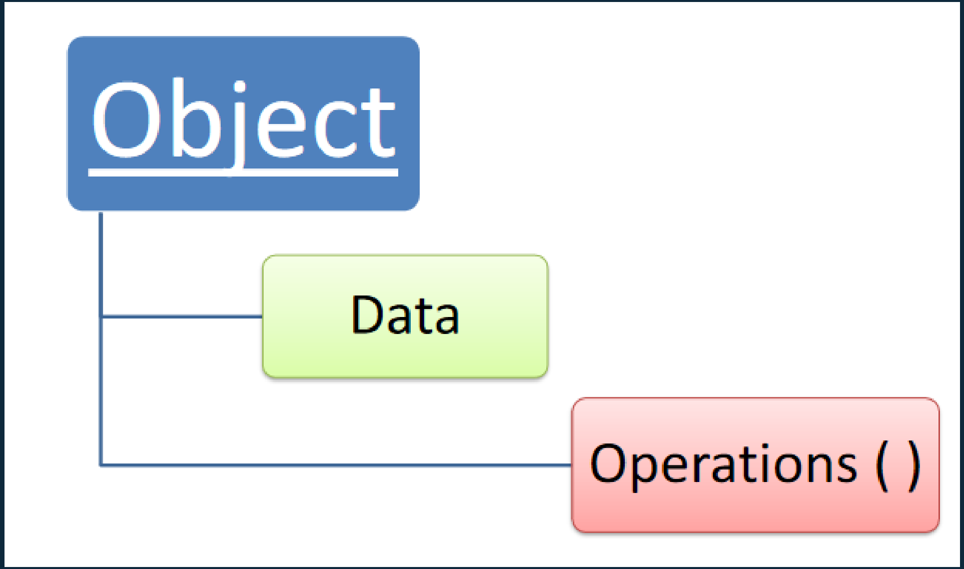
Object

- Object : is a thing (Tangible –Intangible)
- Object Is comprised Of ?



Object

- Object Is comprised Of ?



Object

Data

Operations ( )

What is Class ? Why we need It ?

- Classes: Where Objects Come From–A class is code that describes a particular type of object.
- It specifies the data that an object can hold (the object's fields), and the actions that an object can perform (the object's methods).
- You can think of a class as a code "blueprint" that can be used to create a particular type of object.

Objects and Classes

- When a program is running, it can use the class to create, in memory, as many objects of a specific type as needed.
- Each object that is created from a class is called an instance of the class.

Thanks!

Any questions?