DIFFERENCE BETWEEN PROCEDURE ORIENTED(POP) AND OBJECT ORIENTED PROGRAMMING(OOP)

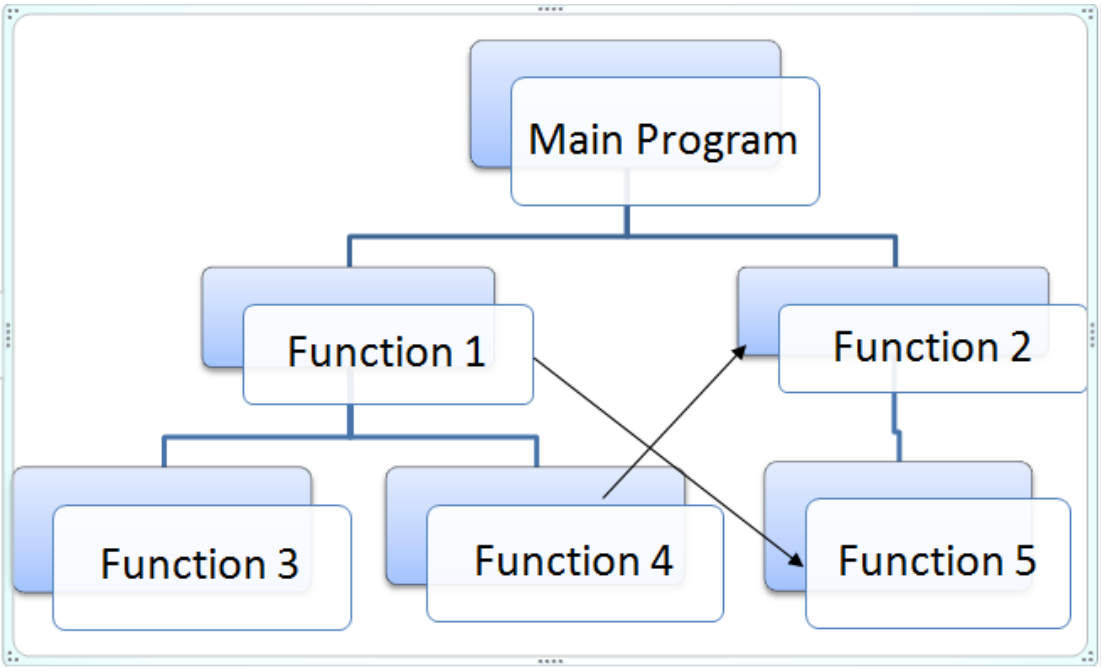
We all know there exist **2** approaches to write a program – **1)** Procedure oriented programming (**POP**) and **2)** Object oriented programming (**OOP**). You can write a program in either way but there are notable differences between both approaches. These 2 approaches are the result of software development evolution over many decades. Since the invention of computer, many approaches and methods have been tried to write a program. It includes methods like**a)** Top-Down programming **b)** Bottom-Up programming **c)** Modular programming **d)** Structured programming etc. The basic purpose or the basic aim of all these methods were same - ***"to make programming efficient"*** – means to make the process of writing a complex program less harder, bug free, easily understandable, easily expandable/modifiable etc

.

In simple words, difference between POP and OOP can be explained like this:- A programmer can handle the task of developing a moderately complex program fairly well enough with POP method. When the program gets even more complex or is of a highly complex type; it would be difficult to write it efficiently using POP method. The whole task of programming would get harder, will take more time, more bugs, more time for debugging etc. In such a case OOP method proves to be much more efficient than POP method. A highly complex program can be developed much efficiently using OOP method. I hope you got a general idea of the difference between both! There is no rule that one should use a particular method to develop a program. Its upto the discretion of the programmer. However in the software development industry, they all follow OOP method as it facilitates collaborative working. The main reason is ***code reusability***. A particular piece of code developed by one programmer can be reused any number of times by any number of other programmers. This makes software development much faster and efficient.

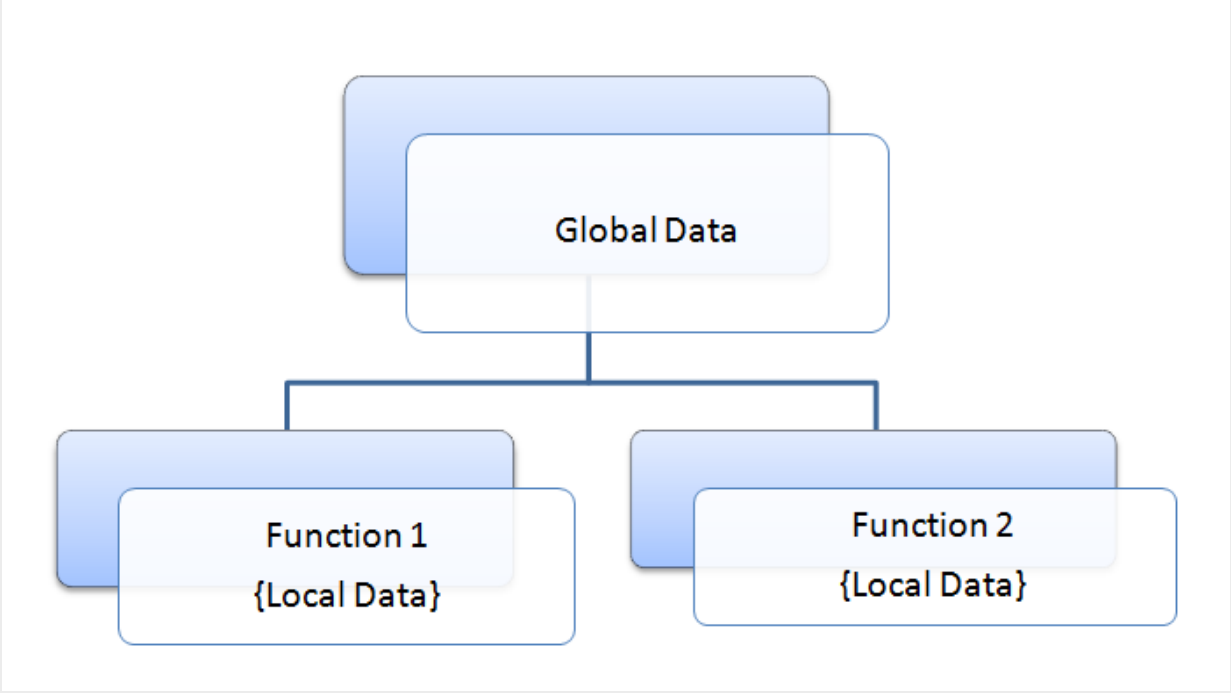
Procedure Oriented Programming

The word "procedure" is the key element here to notice. It means "***a set of procedures***" which is a "***set of subroutines***" or a "***set of functions***". We all know about "functions in C language". **'C'** is a procedure oriented language. In a POP method, emphasis is given to functions or subroutines. Functions are a set of instructions which performs a particular task. Functions are called repeatedly in a program to execute tasks performed by them. For example, a program may involve collecting data from user (reading), performing some kind of calculations on the collected data (calculation), and finally displaying the result to the user when requested (printing). All the 3 tasks of reading, calculating and printing can be written in a program with the help of 3 different functions which performs these 3 different tasks.



Structure of POP method

The problem with POP approach is its handling of data. POP approach gives no importance to data. By **'data'** we mean the information collected from user, the new results obtained after calculations etc. If you are familiar with '**C programming'**, you may recollect **"storage classes"** in C. In C, a data member must be declared **GLOBAL** inorder to make it accessible by 2 or more functions in the program. What happens when 2 or more functions work on the same data member ? If there are 10 functions in a program, all these 10 functions can access a global data member. It is possible one function may accidentally change values of this global data member. If this data member is a key element of the program, any such accidental manipulation will affect the whole program. It will be too difficult to debug & identify which function is causing the problem if the program is really big.



Handling of data & functions in POP

One of the most important feature of C language is "**structures**". If you are familiar with C, recall the way structure is declared in C using keyword **struct**. Structure provides a way to pack together different data types into a single entity. Programmer can pack together integer data, decimal point data (float), array data type etc into a single entity using structure. *Programming using "structure"* was first introduced by C language and this is the single best reason for its wide popularity and acceptance. You know the reason? Structure models real world requirements well into a computer program. The problem with structures was that it handled only data. Structure do not allows to pack together associated functions inside it along with data. All the function to manipulate data members inside structure has to be written separately inside a program. For this reason, a C program has an over dependency on functions.

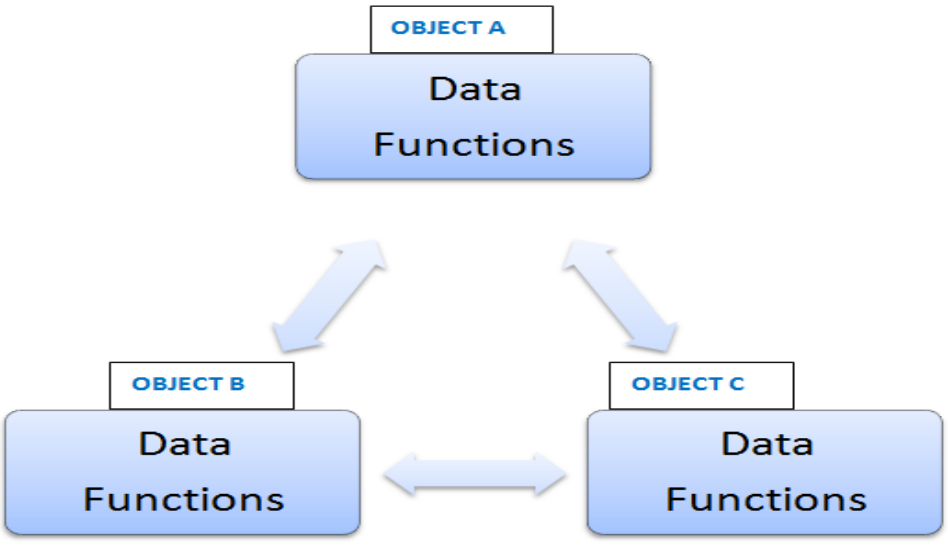
In POP method, a problem is viewed as a sequence of tasks to be implemented like reading, performing calculations, displaying results etc. All the tasks are analysed first and later functions/procedures are developed to implement all these tasks in a program.

Object Oriented Programming

An OOP method differs from POP in its basic approach itself. OOP is developed by retaining all the best features of structured programming method/procedural method, to which they have added many concepts which facilitates efficient programming. Object oriented programming methods brings in

many features and I would say it makes possible an entirely new way of approaching a program. Let's first keep in mind that OOP retains all best features of POP method like functions/sub routines, structure etc.

1) The first feature that any programmer would talk about OOP is "***data hiding***" facility. OOP gives lots of importance to data. Programmer can hide the really important core data from external world using OOP method. The basic concept of OOP revolves around a feature similar to structure in POP, named as **class** in OOP. A class is a feature in OOP which facilitates to pack together different data types along with different functions that manipulate these data members of class. Data members can be declared as **private** or **public** inside a class. To hide the data from external world, a programmer does it by ***declaring the data as private***. Keep note that a class is really similar to***structure in 'C'***. Just like structure, a class packs together different things into a single entity. The major difference between class and structure is with functions. Structure does not allow to pack together data with functions (structure deals with data only) whereas ***class allows to pack together data with its associated functions***. In addition there are differences like 'data hiding' using private/public. Structure does not facilitates data hiding! We know in structure, the structure members are accessed uing an entity called structure variables. In OOP we use another entity named **'object'** to access both data and functions inside a class. We call data or function inside a class as '**class member**'. ***A class member can be accessed from external world (outside the class) using an object of the class only!***



This feature of data hiding is called as "**Data Encapsulation**". Thus one of the major flaws of POP is solved in OOP. OOP ties the data closely to a particular class and its objects. There is no need of "global data types" as in POP and hence data will not flow freely around the program. ***This makes sure there will be no 'accidental modification' of critical data.***

2) Another important feature brought in by OOP is **‘code reusability**‘. This simply means, a piece of code written earlier in a program can be used later. This is made possible by a feature of classes named **“inheritance”.** By using inheritance, one class can acquire the properties of another class. Let me try to explain this using an example. Take the case of a “School Management System”. Initially the management decided to develop a software focused on students only (no data of teachers). The programmer made the software perfectly and while developing he had declared a class dedicated for collecting personal details like Name,Age,Sex,Address etc. After one year school management decides to incorporate data of teachers to the software. The programmer can add this extension within a small time as he can reuse many of the codes he had written earlier by making use of **“inheritance**“. The class personal details is of general nature (Age, Sex etc are same for every person irrespective of student/teacher). Programmer can inherit this class as such to a new one and add extension to this class with new properties like ‘educational qualification’ of the teacher.

Source : http://www.circuitstoday.com/difference-between-procedure-oriented-and-object-oriented-programming