**30/11/2018 CIA-1 Jeevan Koshy (1740256)**

---------------------------------------------------------------------------

***A COMPARATIVE STUDY ON PROCEDURAL ORIENTED PROGRAMMING AND OBJECT-ORIENTED PROGRAMMING WITH APPROPRIATE REAL-WORLD APPLICATIONS***

---------------------------------------------------------------------------

**Report on Procedural Programming**

The term Procedural programming was coined with the view of structured programming in mind. Procedures – also known as methods or functions in a program contain a series of computational steps to be complied and executed. There is hardware support for procedural programming through a stack register to run each of the procedures. In simpler terms, a program has a procedural set of instructions where each statement tells the computer to do something. If at all the program becomes larger – it is divided into functions and each function has a specific task and purpose to be carried out. A few characteristics of procedural programming are –

* It emphasises on processes rather than data.
* Data moves openly around the system
* Most of the functions share the data globally.
* Contains a set of instructions such as – reading, calculating, printing and so on……to solve a problem.
* The program is divided into different functions to execute many tasks.  
  Some drawbacks of procedural programming are –
* Data is secondary although data is the main reason the program exists.
* Since the variables are global in the functions used, the data can get corrupted.

**Report on Object Oriented Programming**

An **Object** is a real-world entity such as a pen, chair, table, computer, watch, etc. **Object-Oriented Programming** is a methodology or paradigm that is used to design a program using classes and objects.

A class is a collection of data and methods.

Object-oriented programming (OOP) is a programming language model organized around [objects](https://searchmicroservices.techtarget.com/definition/object) rather than "actions" and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.

The main aim of OOP is to bind together the data and the functions that operates on them so that no other part of code can access this data except that function.

Object-oriented programming takes the view that what we really care about are the objects we want to manipulate rather than the logic required to manipulate them.

It simplifies the software development and maintenance by providing some concepts such as:

* [Object](https://www.javatpoint.com/object-and-class-in-java) – an entity that has a state or behaviour
* Class – a collection of objects
* [Inheritance](https://www.javatpoint.com/inheritance-in-java) – one object acquires behaviour of parent object
* [Polymorphism](https://www.javatpoint.com/runtime-polymorphism-in-java) – one task is performed in different ways
* [Abstraction](https://www.javatpoint.com/abstract-class-in-java) – hiding internal details but showing functionality
* [Encapsulation](https://www.javatpoint.com/encapsulation) – Binding and wrapping code in a single unit.

It allows users create the objects that they want and then create methods to handle those objects. Manipulating these objects to get results is the goal of Object - Oriented Programming.

**Comparison**

|  |  |
| --- | --- |
| **PROCEDURAL ORIENTED** | **OBJECT ORIENTED** |
| Divided into functions | Divided into objects |
| Importance given to actions | Importance given to data |
| Top down approach | Bottom up approach |
| No access specifier | Public, private, protected |
| Data can move freely | Communicates through functions |
| Uses global data | Public or private variables |
| Less secure | More secure |
| Overloading is not possible | Function overloading is possible |
| Eg. – C, Pascal | Eg. – C++, Java |

**Real World Problem**

**Single Level Inheritance –**

MOBILE

SAMSUNG

**Multi-Level Inheritance –**

MOBILE

SAMSUNG

SAMSUNG S5

**Multiple Inheritance –**

ITELEPHONE

IMOBILE

SAMSUNG

**References -**

<http://www.sciencehq.com/computing-technology/procedural-oriented-programming.html>

<https://en.wikipedia.org/wiki/Procedural_programming>

<https://www.javatpoint.com/java-oops-concepts>

<https://freefeast.info/general-it-articles/difference-between-procedure-oriented-programming-and-object-oriented-programming-procedure-oriented-programming-vs-object-oriented-programming/>

---------------------------------------------------------------------------