**Assignment No 01**

**Object Oriented Programming (OOP):**

Object-Oriented Programming (OOP) is a programming technique that operates on the notion of classes and objects. In (OOP) such fragments of code are generated that are usable repeatedly. An arrangement in the form of blueprints (classes) is generated that is used in addition to forming individual instances of entities.

**Procedural Programming (POP):**

Procedural language is kind of a structured programming language that uses the concepts of functions. Functions, also called procedures basically consist of a series of computational tasks to be carried out once the function is called.

**Advantages of OOP over POP:**

1. **Simplicity:**

OOP is much more suited for a large project. Which basically means that if there is a certain area of project that a team of developers need to work on, they can easily do that, against the procedural programming, in which the team would’ve needed to understand the entire program from start to end, in order to work on their own part.

1. **Security:**

Procedural programming uses the concept of global variable. Hence, there is no proper way of hiding/encrypting the data. Whereas in OOP, data can be made much more secure by using concepts such as **encapsulation** and using specifiers such as private, public, protected etc. Contrary to OOP, the global variables in procedural programming are accessible to any function from the entire programming which leads to flaws in the final results as the data is not restricted.

1. **Maintenance:**

Modifying existing codes and troubleshooting is much easier in OOP as compared to procedural programming. Hence making OOP programs much more maintainable.

1. **Reusability:**

In OOP, the codes are highly re-usable. As each module is independent. One module can be used in other modules.

**Advantages of POP over OOP:**

1. **Smaller Projects:**

Smaller projects which are less complex are more suited for procedural programming as the concept of objects becomes too complex even for a simple small program.

1. **Efficiency:**

Programs written according to OOP are overly generalized and take up considerable amount of time to run in memory. These programs also take considerable storage whereas programs written in procedural programming paradigm occupies relatively low space on memory.

1. **Learning Difficulty:**

Object Oriented Programming and its languages can be a little more difficult to learn as compared to procedural languages. The learning curve for OOP can be a little steep for many people.

**Examples:**

1. For example, we want to **store the data of all employees** in a company. So, for this purpose using procedural paradigm will not be a great idea as we would need to write code for every employee which will make our program less efficient. So, to make our code reusable we will take help of object-oriented programming where we will just define all the fields and methods of class once and use them for unlimited number of employees.
2. Suppose you are the owner of a **car showroom** and you want to keep track of all the sales of your cars and want to know all the data that how many models of a particular car are left, etc. So, in this example the use of procedural oriented program will not be fruitful as a showroom contains cars of numerous models and to keep track the record of each and every car is a hectic task. Here comes the use of OOP where we can categorize the cars in limited sections and make our task easier.
3. A **system for storing CNIC** can also be a good example for OOP paradigm. Because firstly, there are a large number of people (objects) but there can be more than one person with the same name. But because this is done in OOP, each object has its own attributes and methods hence removing the issue of having more than one person with the same name. This system is also not practically suitable for procedural paradigm because every person has its own attributes.