

Object-oriented programming (OOP) is a computer programming model that plans software design based on data, or objects, versus functions and logic. In this project the objects could be specified as a data field that contain exclusive attributes and behavior.

Procedural .....

In this project, OOP targeted the objects that the user aimed to manipulate versus the logic needed to manipulate them. This style of programming is especially adaptable when the programs are large, complex and constantly updated or maintained. This often involves programmes for manufacturing and design, in addition to mobile applications.

The execution of an object-oriented programme style in this project meant the mode was highly usable, in the collaborative development. Other advantages of OOP included code reusability, scalability and efficiency.

The strength of the OOP approach is the data modelling that it offers the developer whereby the objects could be easily manipulated and identified in how they interact with one another.

An example of an object in the project is the student object as a physical entity, this is a human being who is identified by properties like name, another real world example is a computer programme, such as widgets.

When an object is given an identity, it is given a class label that specifies the kind of data and all logic sequences that can manipulate it. Each sequence or step is known as a method. Objects can interact and communicate with interfaces known as messages.

In object oriented programming, a programme is broken into smaller parts known as objects. Procedural programming does not have any truly established way to hide data, thus it is less secure. Object oriented programming offers data hiding, so, it is a more secure way to store data in the project.

In procedural programming, the programme is divided into small parts known as functions. In object oriented programming, the programme is broken into lesser parts called objects. Procedural programming follows a top down approach, however, Object oriented programming is quite the opposite. There is no access specifier in procedural programming. Object oriented programming, contain access specifiers like private, public, protected etc. Adding new data and function is easy in OOP, the addition of new data and function is not as easy in procedural. In procedural programming, overloading is not possible but overloading is offerable in object oriented programming. In

procedural programming, function is more important than data. In object oriented programming, data is more important than function. Procedural programming is based

on an unreal world. Examples: C, FORTRAN, Pascal, Basic etc. Object oriented programming is based on real world. Examples: C++, Java, Python, C# etc.

There are many advantages to hiding information regarding attributes and methods using encapsulation in OOP programming. It does not allow other users from writing scripts or APIs that utilise your code. With encapsulation, users of a class do not know how a class contains its data and the owner can change the data type of a field without insisting that users of the class modify their code. IT organizations can enforce encapsulation as a way to guard sensitive data and retain compliance with industry-specific data security and privacy. The encapsulation strategy aids in compartmentalising data, reducing vulnerabilities by allowing users only that information on code that is necessary and specific.

Inheritance in object oriented programming has many huge benefits. In this project, the user defined a new class Student class with little or no modification to an existing class. The new class, often noted as the child class and the one from which it inherits is called the parent class. New features were added to it when required. This allowed the re-usability of code.

In the procedural approach, data was allowed to move freely, function is more important than data in this approach. The solutions in this style were most often found to be concise and simple. The user found the codes to have the option to be reused in many parts of the programme. The programming paradigm in particular utilised less memory on the computer. It is easier for logging the direction of the codes in the programme executed in the procedural programming paradigm. This style of programming had the support of dividing the functionalities into a number of procedures. The user found that the code could more easily be broken down into small, manageable parts. The approach permitted a sequence of instructions to be executable, each of which tells the computer to carry out an action, such as taking inputs from the user, doing calculation, and displaying output. However, these functions do not permit code duplication owing to the same function call from multiple locations. This paradigm is only a utility for medium sized software applications.

## References

<https://www.geeksforgeeks.org/differences-between-procedural-and-object-oriented-programming/>

<https://searchapparchitecture.techtarget.com/definition/object-oriented-programming-OP>

<https://www.sumologic.com/glossary/encapsulation/>

<https://www.tuannguyen.tech/2019/05/programming-paradigms-what-is-procedural-programming/>

<https://www.tuannguyen.tech/2019/05/programming-paradigms-what-is-procedural-programming/>

