|  |  |
| --- | --- |
| **Procedural Programming** | **Object-oriented Programming** |
| It focuses on the process and functions. | It focuses on the data and classes. |
| It is not easy to maintain. | It is easy to maintain. |
| In this paradigm, if a sub-procedure has to be modified, it becomes difficult to find and maintain it. | In this paradigm, it is easy to maintain code and modify existing code. |
| Due to its complexity, development time increases. | Due to easy maintenance, development time reduces. |
| Procedural programming languages are not as faster as object-oriented. | The object-oriented programming languages are faster and more effective. |
| Procedural uses procedures, modules, procedure calls. | Object-oriented uses objects, classes, messages. |
| It focuses on procedure rather data which has priority in data-driven systems. | It focuses on data rather than procedures. |
| In procedural programming, designs cannot be reused and recycled throughout the program. | In object-oriented programming, designs can be reused throughout the program. |
| While solving issues in procedural programming, issues need to be addressed individually. | In object-oriented programming, objects and classes can be referenced throughout the program. |

|  |  |
| --- | --- |
| Data hiding is not possible. | Data hiding is possible, hence more secure than procedural. |
| Operator overloading is not allowed. | In OOP, operator overloading is allowed. |
| It has no such concepts like Inheritance. | It has four main concepts – Abstraction, Encapsulation, Inheritance, and Polymorphism. |
| The different parts of the program are connected via parameter passing. | The different functions of objects are connected via message passing. |

**Based on Characteristics**: Procedural programming has Local variables, sequence, selection, iteration, and modularisation. Object-oriented programming has Objects, methods, message passing, information hiding, data abstraction, encapsulation, polymorphism, inheritance, serialisation-marshalling.

**Accessing modes**: In Object-oriented programming, there are three accessing modes – Public, Private, and Protected. There are no such access modes in Procedural programming.

**Execution**: In Object-oriented programming, various functions can execute simultaneously. In procedural programming, there is a systematic approach in which functions get executed step-by-step.

**Data Control**: In Object-oriented programming, data and functions are accessible within the same class while in procedural programming, data can move freely.

**Security**: Object-oriented programming is more secure than procedural programming, because of the level of abstraction or we can say data hiding property. It limits the access of data to the member functions of the same class. While there is no such data hiding in the procedural programming paradigm.

**Process**: Object-Oriented programming follows the**bottom-up approach**while Procedural programming follows the **top-down approach** while designing a program.

**Division**: In Object-oriented programming, the program is divided into small entities called objects whereas in Procedural programming the program is divided into sub-procedures.