**Differences between object oriented programming language and procedural programming language:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no.** | **On the basis of** | **Procedural Programming** | **Object-oriented programming** |
| **1.** | **Definition** | It is a programming language that is derived from structure programming and based upon the concept of calling procedures. It follows a step-by-step approach in order to break down a task into a set of variables and routines via a sequence of instructions. | Object-oriented programming is a computer programming design philosophy or methodology that organizes/ models software design around data or objects rather than functions and logic. |
| **2.** | **Security** | It is less secure than OOPs. | Data hiding is possible in object-oriented programming due to abstraction. So, it is more secure than procedural programming. |
| **3.** | **Approach** | It follows a top-down approach. | It follows a bottom-up approach. |
| **4.** | **Data movement** | In procedural programming, data moves freely within the system from one function to another. | In OOP, objects can move and communicate with each other via member functions. |
| **5.** | **Orientation** | It is structure/procedure-oriented. | It is object-oriented. |
| **6.** | **Access modifiers** | There are no access modifiers in procedural programming. | The access modifiers in OOP are named as private, public, and protected. |
| **7.** | **Inheritance** | Procedural programming does not have the concept of inheritance. | There is a feature of inheritance in object-oriented programming. |
| **8.** | **Code reusability** | There is no code reusability present in procedural programming. | It offers code reusability by using the feature of inheritance. |
| **9.** | **Overloading** | Overloading is not possible in procedural programming. | In OOP, there is a concept of function overloading and operator overloading. |
| **10.** | **Importance** | It gives importance to functions over data. | It gives importance to data over functions. |
| **11.** | **Virtual class** | In procedural programming, there are no virtual classes. | In OOP, there is an appearance of virtual classes in inheritance. |
| **12.** | **Complex problems** | It is not appropriate for complex problems. | It is appropriate for complex problems. |
| **13.** | **Data hiding** | There is not any proper way for data hiding. | There is a possibility of data hiding. |
| **14.** | **Program division** | In Procedural programming, a program is divided into small programs that are referred to as functions. | In OOP, a program is divided into small parts that are referred to as objects. |
| **15.** | **Examples** | Examples of Procedural programming include C, Fortran, Pascal, and VB. | The examples of object-oriented programming are - .NET, C#, Python, Java, VB.NET, and C++. |