|  |  |  |
| --- | --- | --- |
| **Differentiate between Structured Programming and Object-Oriented Programming** | | |
| **Comparison Criteria** | **Structured Programming** | **Object-Oriented Programming** |
| Focusing on | Processes -> Logical Structures -> data required for that process. | Data |
| Design approach | Top-down approach | Bottom-up approach |
| Also known as | Modular Programming  Stepwise refinement  Top-down design | Supports inheritance, encapsulation, abstraction, polymorphism, interfaces, etc |
| Program | Divided into small self-contained functions | Divided into small entities called objects |
| Security | Less secure as there is no way of data hiding. | More secure as having data hiding features. |
| Used to solve | Moderately complex programs | Any complex program. |
| Reusability | Less reusability and more function dependency | More reusability and less function dependency |
| Level of abstraction | Less flexibility | More flexibility |
| Written codes and how they are processed | All methods / functions are written globally and run sequentially | With the use of objects, call to methods of a particular objects is more dynamic, giving the benefit of source code reusability and high maintainability. |