Difference between object-oriented and object-based language.

Ans:

|  |  |  |
| --- | --- | --- |
| Aspect | Object-Oriented Language | Object-based Language |
| Defination | Object-oriented languages fully support all the features of object-oriented programming, including classes, objects, inheritance, polymorphism, and encapsulation. | Object-based languages primarily  support objects and object-related  concepts but may lack certain key features like classes or inheritance. |
| Class | Typically, object-oriented languages use classes to define object blueprints or templates. Classes can include both data (attributes) and methods (functions). | Object-based languages may not have classes or may use prototypes or other mechanisms instead of classes to define objects. |
| Inheritance | Object-oriented languages often support inheritance, allowing new classes (subclasses) to be derived from existing classes (superclasses). Subclasses inherit attributes and methods from superclasses. | Object-based languages may support a limited form of inheritance or none at all. Inheritance may not include methods or may be based on object cloning or prototyping. |
| Polymorphism | Object-oriented languages support polymorphism, allowing objects of different classes to respond to the same method name in a way that is appropriate for their specific class. This is achieved through method overriding and dynamic binding. | Object-based languages may support a basic form of polymorphism, but it may be limited or less flexible compared to object-oriented languages. |
| Encapsulation | Object-oriented languages support encapsulation, which hides the internal details of objects and restricts access to data through access modifiers like public, private, and protected. | Object-based languages may support encapsulation to a lesser  extent or rely on fewer access modifiers. |
| Use Cases | Object-oriented languages are suitable for building complex, structured, and modular software systems. They provide a strong foundation for designing and implementing large-scale applications. | Object-based languages are often  used for scripting, web development, and applications where simplicity, flexibility, and  dynamic behavior are more important than strict class-based  structures. |