|  |  |
| --- | --- |
| Polymorphism | 1 |
| Overriding | 1 |
| Overloading | 1 |
| Packages | 2 |

**Polymorphism:**

* Polymorphism in object-oriented programming (OOP) is a concept where objects of different classes can be treated as objects of a common superclass.
* This allows a single interface to be used to manipulate objects of different types, providing flexibility and extensibility in code design.
* It allows developers to write more efficient code and redefine methods for derived classes, making it a must-learn concept in OOP

**Overriding:**

* It refers to the ability of a subclass to provide a specific implementation of a method that is already provided by its parent class.
* In method overriding, the method in the subclass has the same name, same parameters or signature, and the same return type
* It overrides the base class and executes the derived class
* Method overriding is a fundamental concept in OOP and is supported by languages such as Java, C++, and Perl

**Overloading:**

* It refers to the ability of a class to have multiple methods with the same name but different parameters
* This allows developers to perform different operations based on the type and number of arguments passed to the method
* Overloading provides several benefits, including cleaner code, reduced redundancy, and improved readability.

**Packages:**

* A package is a namespace that organizes a set of related classes and interfaces
* Packages are used to avoid naming conflicts, write better maintainable code, and provide controlled access
* Packages are named in reverse order of domain names, and adding a class to a package is done by using the package name at the top of the program