# Object-Oriented Programming Concepts

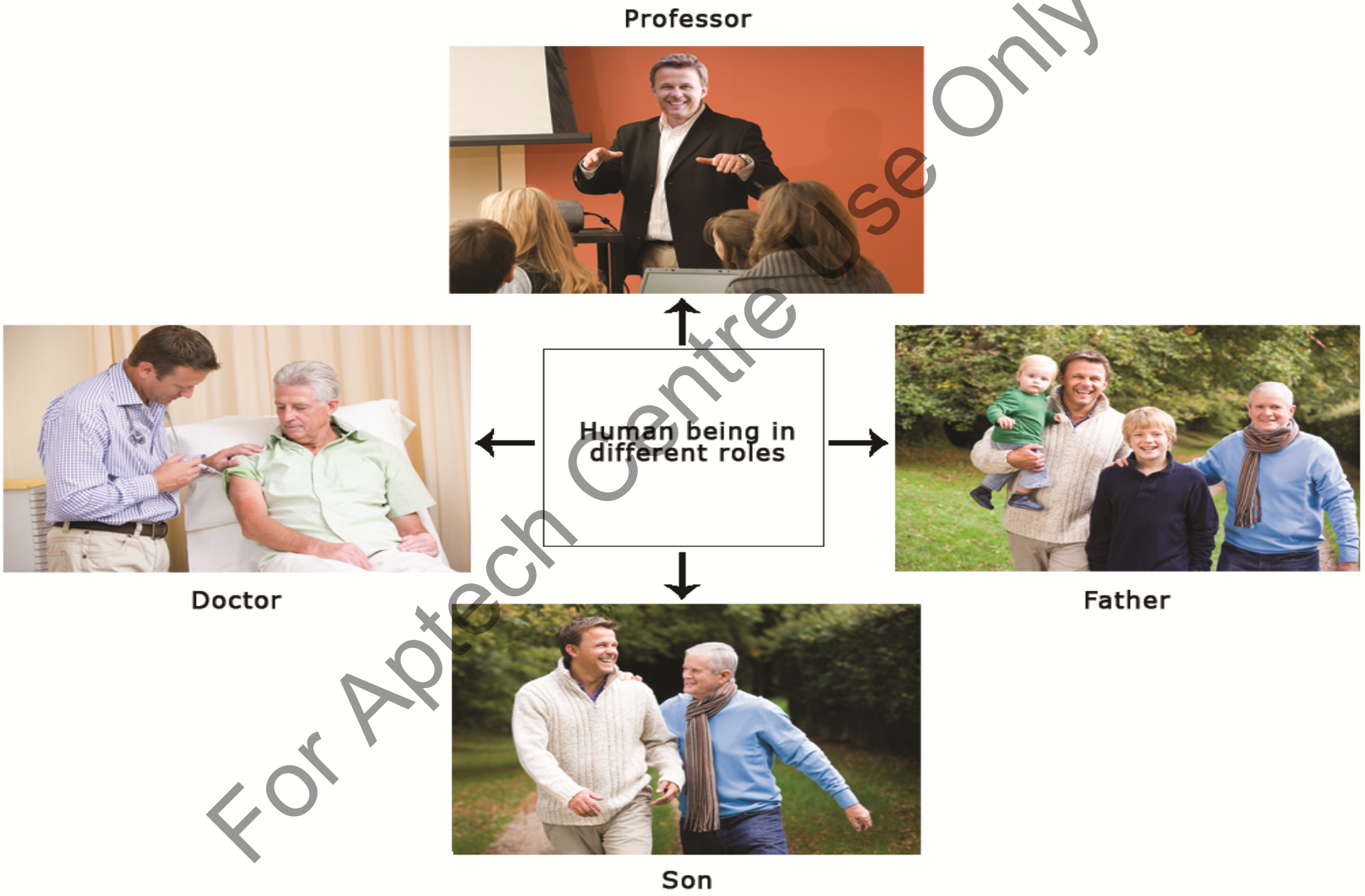
Session: 6

## Polymorphism

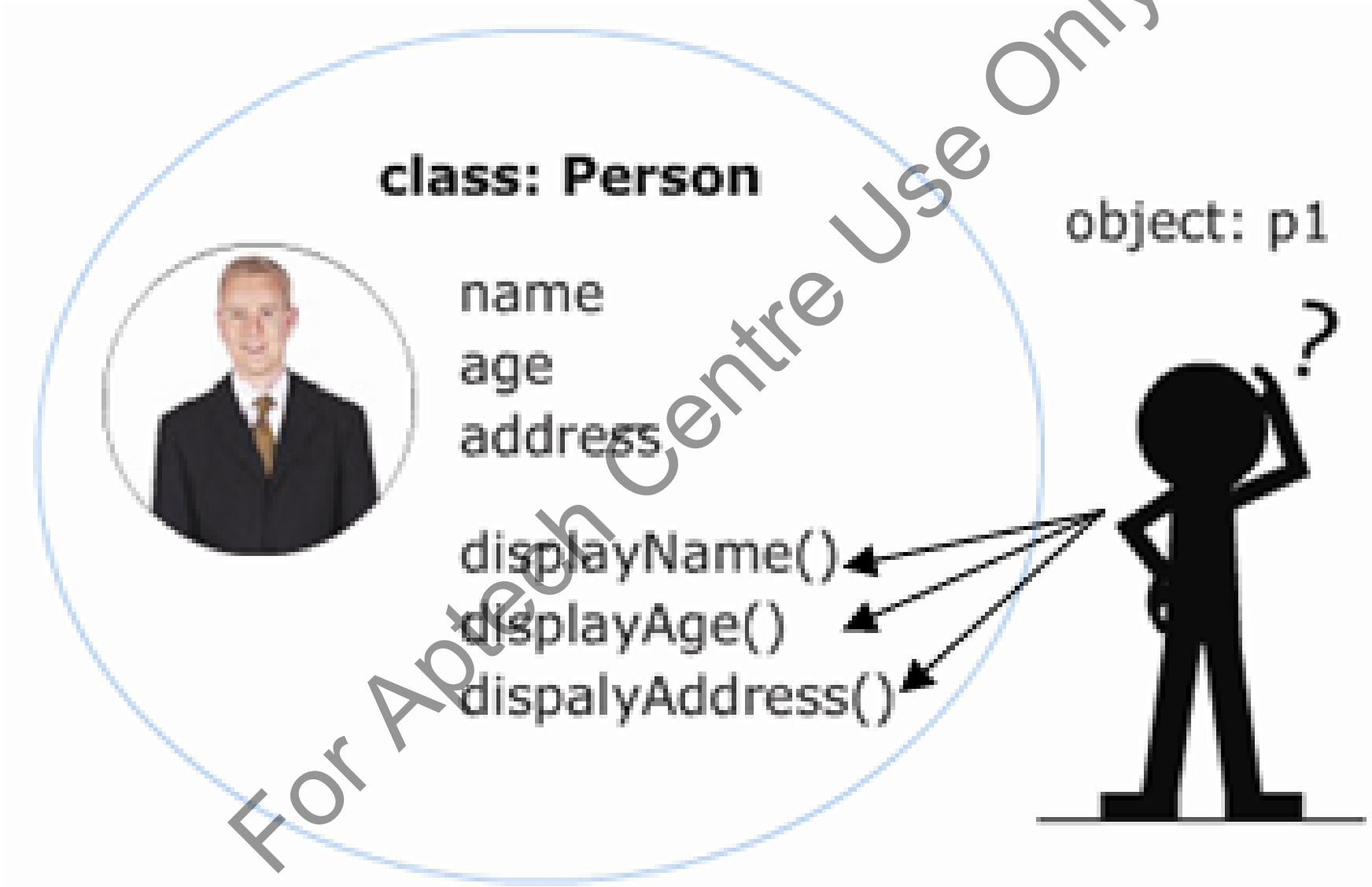
# Objectives

- Explain Polymorphism
- List the different forms of Polymorphism
- Define Overloading and Overriding
- Define Polymorphic variable and Generics
- Explain Static and Dynamic Polymorphism

# Polymorphism

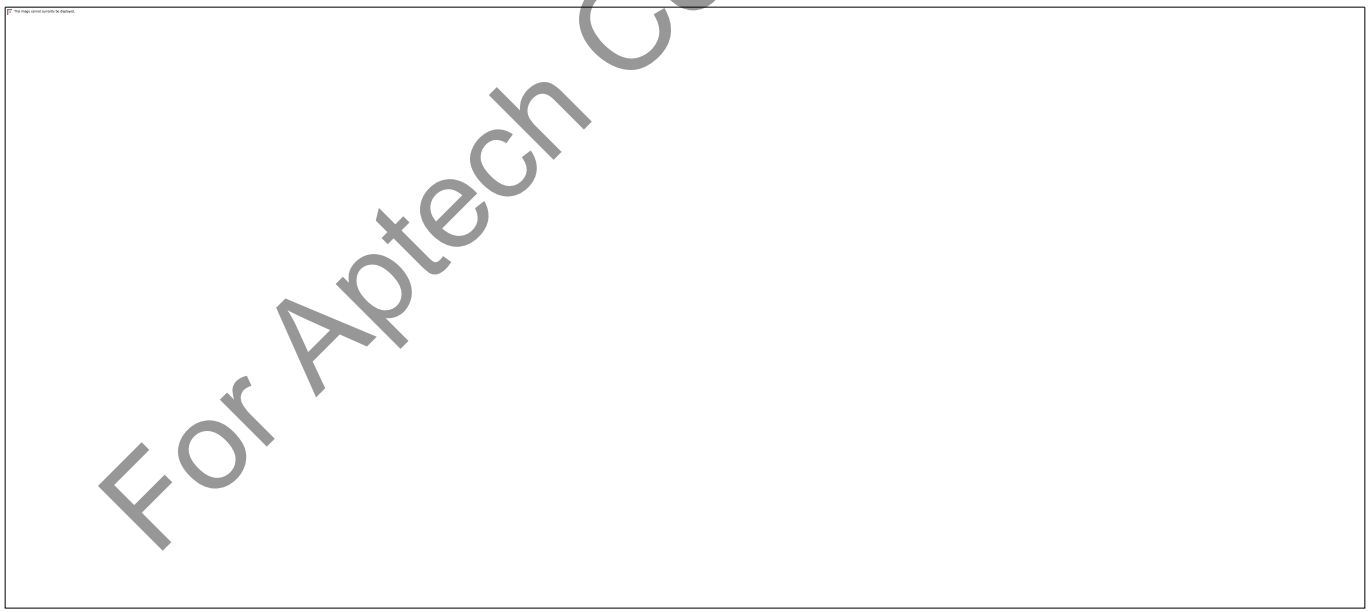


# Overloading



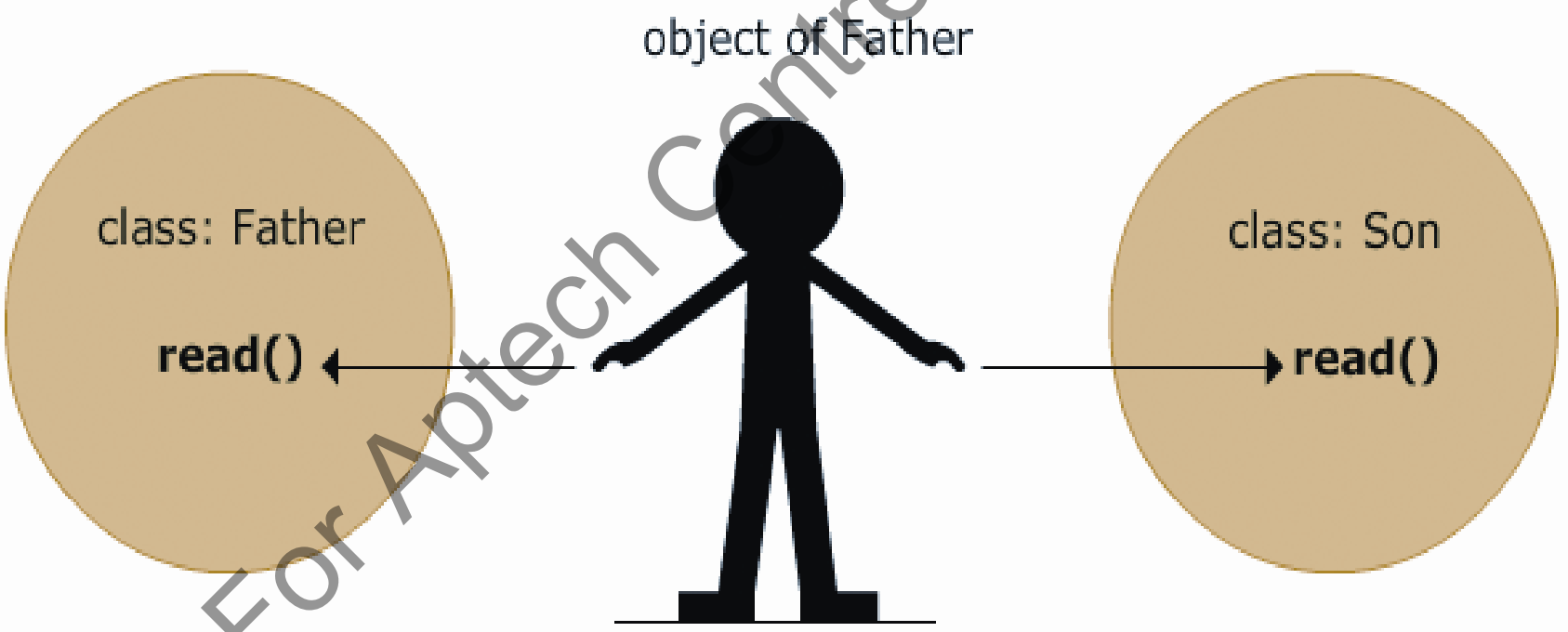
# Overriding

- Child class can override the method of a parent class
- Change the implementation of the method to suit its own requirement
- The figure shows the child implements the command in his own way disregarding what his parent has said.

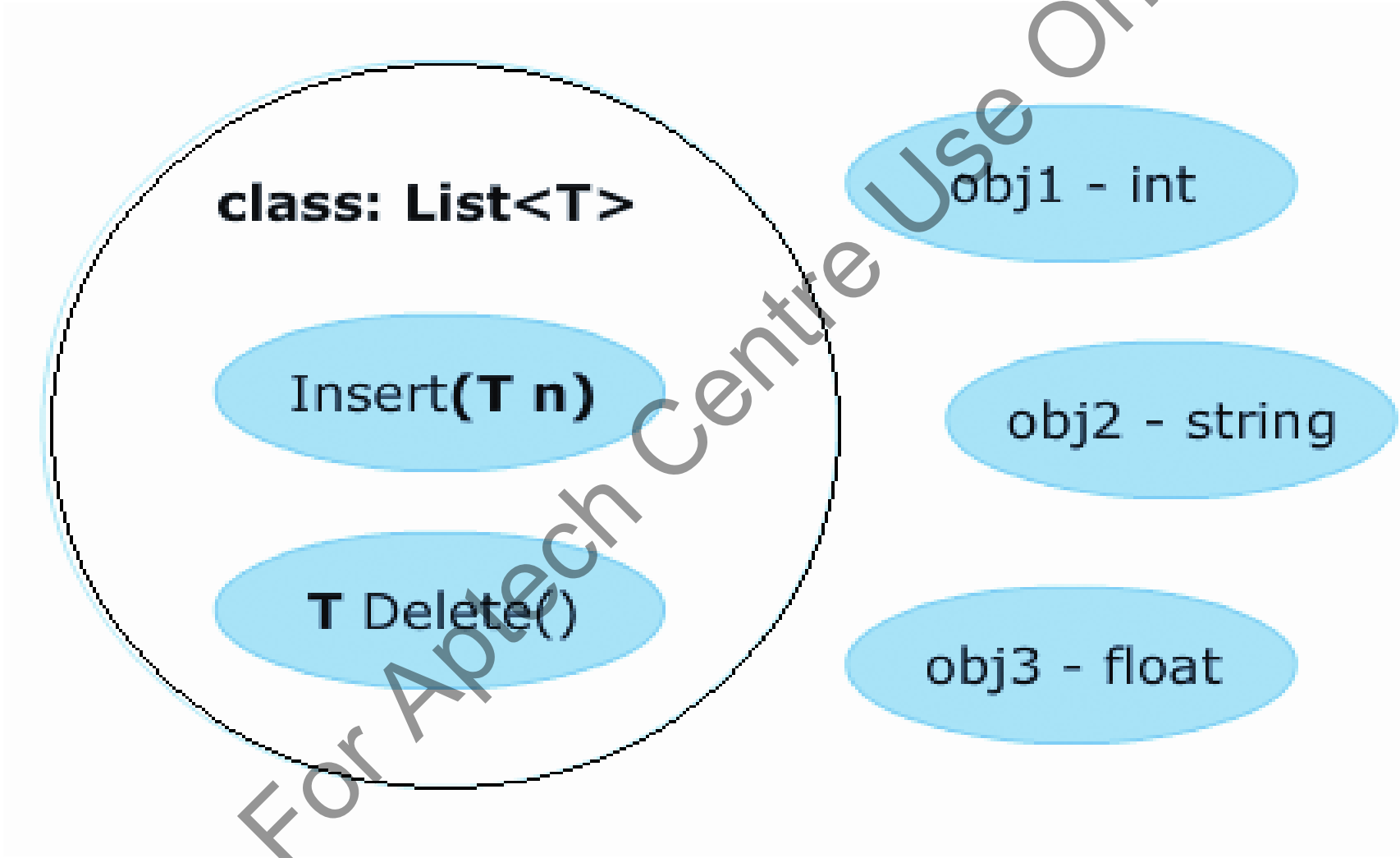


# Polymorphic Variable

Technique by which an object of parent class can be assigned reference to a child class



# Generics



# Static and Dynamic Polymorphism

It is based on the time at which the methods and variables are bound to their calls

Static polymorphism is a technique wherein the method calls and data bindings are fixed at compile time

Dynamic polymorphism is a technique wherein the method calls and data bindings are not fixed at compile time and are decided at run time

# Summary 1-2

- Polymorphism is a technique that allows an entity such as a variable, a method, or an object to have more than one form.
- Method Overloading is a technique in which a method with the same name can have several implementations by changing its signature.
- Method Overriding is a technique where a child class can change the implementation of the method inherited from the parent class to suit its own requirement.

# Summary 2-2

- An object that is declared as one type but holds a value of a different type is known as a polymorphic variable.
- A generic function or variable is one whose type is not fixed during compilation but is decided by the value passed by the user at runtime.
- Static polymorphism is a technique wherein the function calls and data bindings are fixed at compile time.
- Dynamic polymorphism is a technique wherein the function calls and data bindings are not fixed at compile time and are decided at run time.