FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Inheritance

CSPROG2
Computer Programming 2 for CS

Specific Objective

- Learn about the definition of inheritance
- Discuss the different benefits of inheritance in the OOP
- Differentiate the function of superclass and subclass in the inheritance
- Learn the different types of inheritance

Inheritance

The technique of deriving new class definitions from an existing class definition.

Inheritance

The following are the benefits of using class inheritance in OOP:

- Re-use of predefined and well-tested classes
- Standardization of behaviors across a group of classes
- Ability to use members of a family of classes interchangeably in methods

Inheritance

Superclasses and Subclasses

- Superclass is the class from which another class inherits properties. This is a class that is on top of a hierarchy.

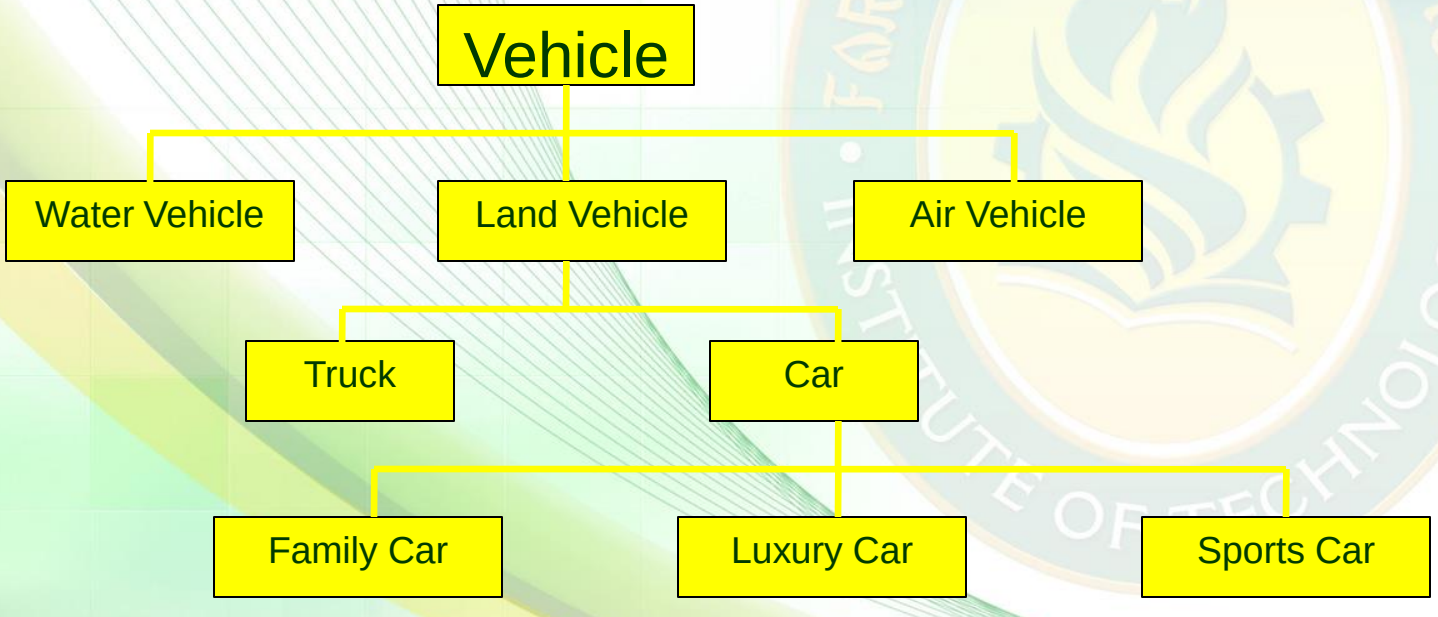
Inheritance

- **Superclasses and Subclasses**

➢ **Subclass is a class that inherits all the non-private attributes and methods, except constructors from a superclass. This class has the ability to override methods of the superclass.**

Inheritance

- **Vehicle Class Hierarchy**



Inheritance

Syntax for Implementing Inheritance:

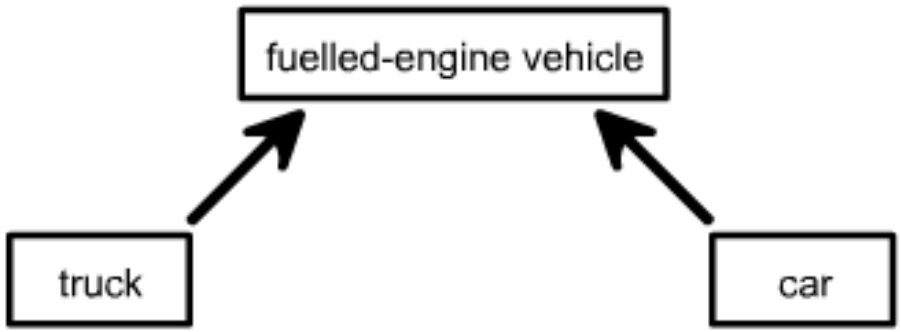
```
public class Subclass    extends SuperClass
{
    // attributes or data declarations

    //constructor and methods definitions
}
```

Inheritance

- **Types of Inheritance:**

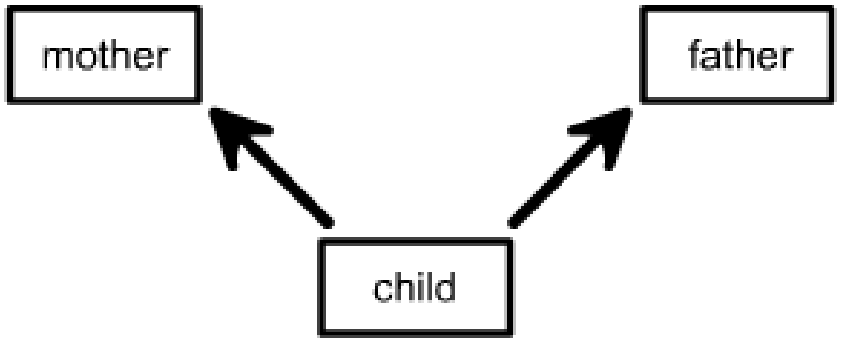
1. **single inheritance** - subclasses are derived from a single superclass. The derived class inherits the data members and methods of the superclass.



Inheritance

- **Types of Inheritance:**

2. **multiple inheritance** - subclasses are derived from more than one superclass. The derived class inherits the data members and methods its superclasses.



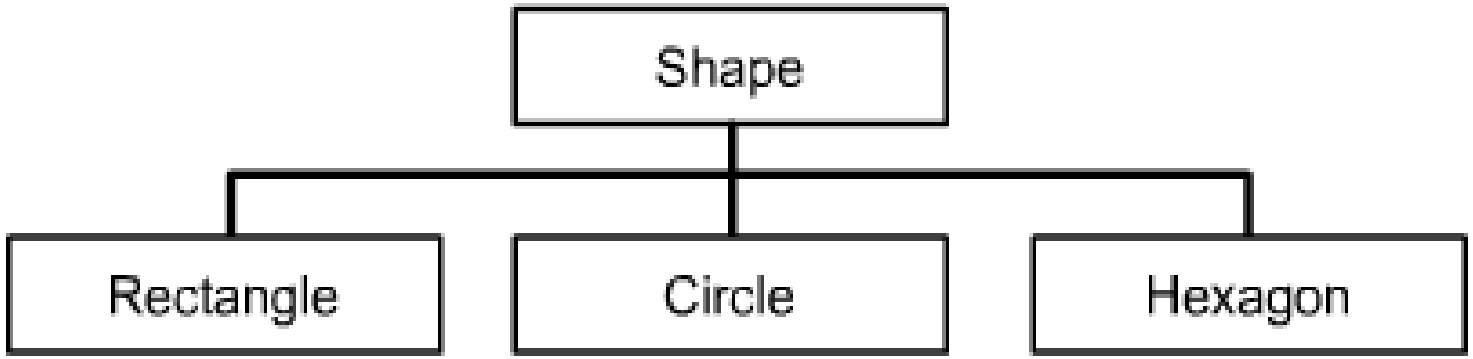
Inheritance

- **The Abstract Class and Interface:**

➢ **Abstract Class** – contains one or more abstract methods and, therefore, can never be instantiated. It is defined so that other classes can extend them and make them concrete by implementing the abstract methods.

Inheritance

abstract class



Inheritance

- **Syntax for Declaring Abstract Class:**
-

```
abstract class ClassName    {
    // abstract method declarations
    public abstract returnType methodName(ArgsList);
}
```

Inheritance

- **The Abstract Class and Interface:**
-

➢ **Interface –** an abstract class that represents a collection of method definitions and constant values. It can later be implemented by classes that define the interface using the implements keyword.

Inheritance

- **Question & Answer**