

INTRODUCTION

This chapter begins with brief discussion of various Programming paradigms that C++ supports *i.e., procedural programming style, object based programming style and object oriented programming style.* This is followed by detailed discussion of object oriented programming and its implementation. All this shall help you appreciate OOP style and its advantages over other programming methodologies.

What is a Programming Paradigm

A Programming Paradigm defines the methodology of designing and implementing programs using the key features and building blocks of a programming language.

Procedural Programming

Procedural programming paradigm separates the functions and the data manipulated by them. This leads to many problems when it comes to extending the software or maintaining the software.

- *Procedural Programming is susceptible to design changes.*
- *Procedural Programming leads to increase time and cost overheads during design changes.*

Object Based Programming

Object based programming is a newer paradigm that implements some features of object oriented programming but not all. In *object based programming*, data and its associated meaningful functions are enclosed in one single entity a *class*. Classes enforce information hiding and abstraction thereby separating the implementation details and the uses interface.

[For instance, you can consider a calculator, its interface would include a small display screen and a few buttons that is available to the user and implementation details *i.e.* how actual calculations are taking place are hidden from user.]

- *Object based programming localizes the implementation details.*
- *Object based programming is subject of Object Oriented programming.*

Advantages of Object Based Programming

- i. It overcomes most shortcomings of procedural programming,
- ii. It localizes changes and hides implementation details from user,
- iii. It supports user-defined types,
- iv. Implements information hiding and abstraction *etc.*

Limitations

However, object based programming suffers from a major **limitation** and that is its inability to represent real world relationships that exist among object.

For example, both car and truck are vehicles. This can not be represented in object based programming as it does not support inheritance.

Object Oriented Programming

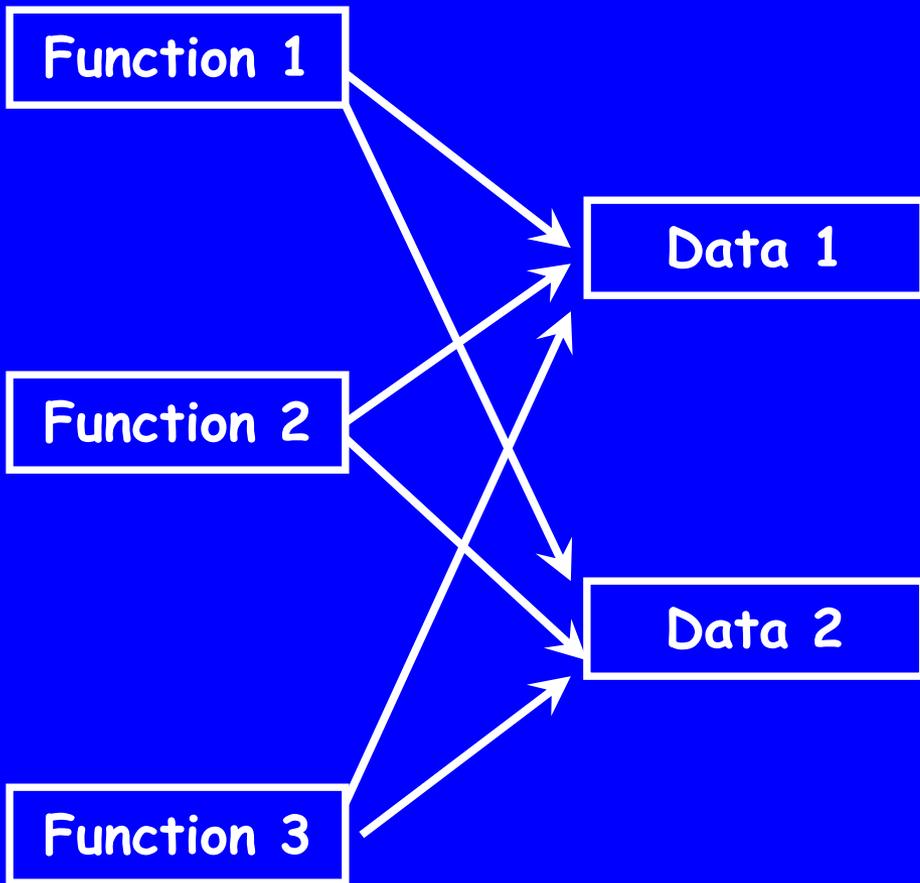
The *object oriented programming* paradigm is superset of object based programming. It offers all the features of *object based programming* and overcomes its limitation by implementing **inheritance**.

What is an **OBJECT** ?

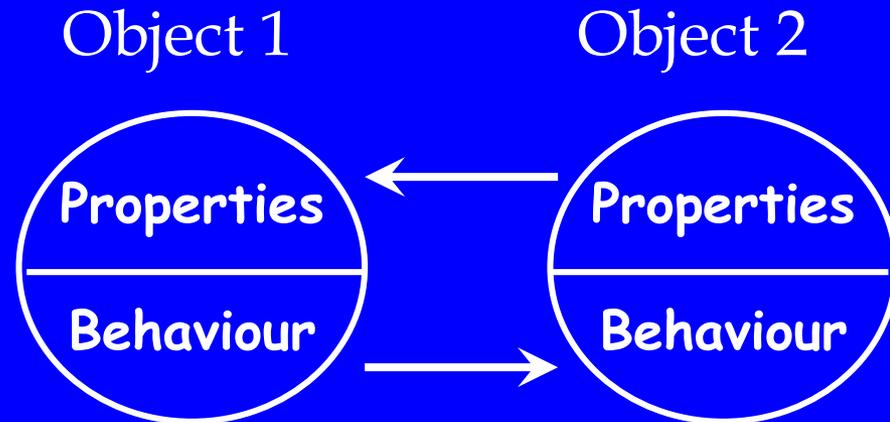
Object is an identifiable entity with some characteristics and behaviour.

What is a **CLASS** ?

A *class* is a group of objects that share common properties and relationships.



All data is openly available to all functions in the program



Data and functions enclosed within objects. New objects communicate with one another.

BASIC CONCEPTS OF **OOP**

These general concepts of *OOP* are given below:

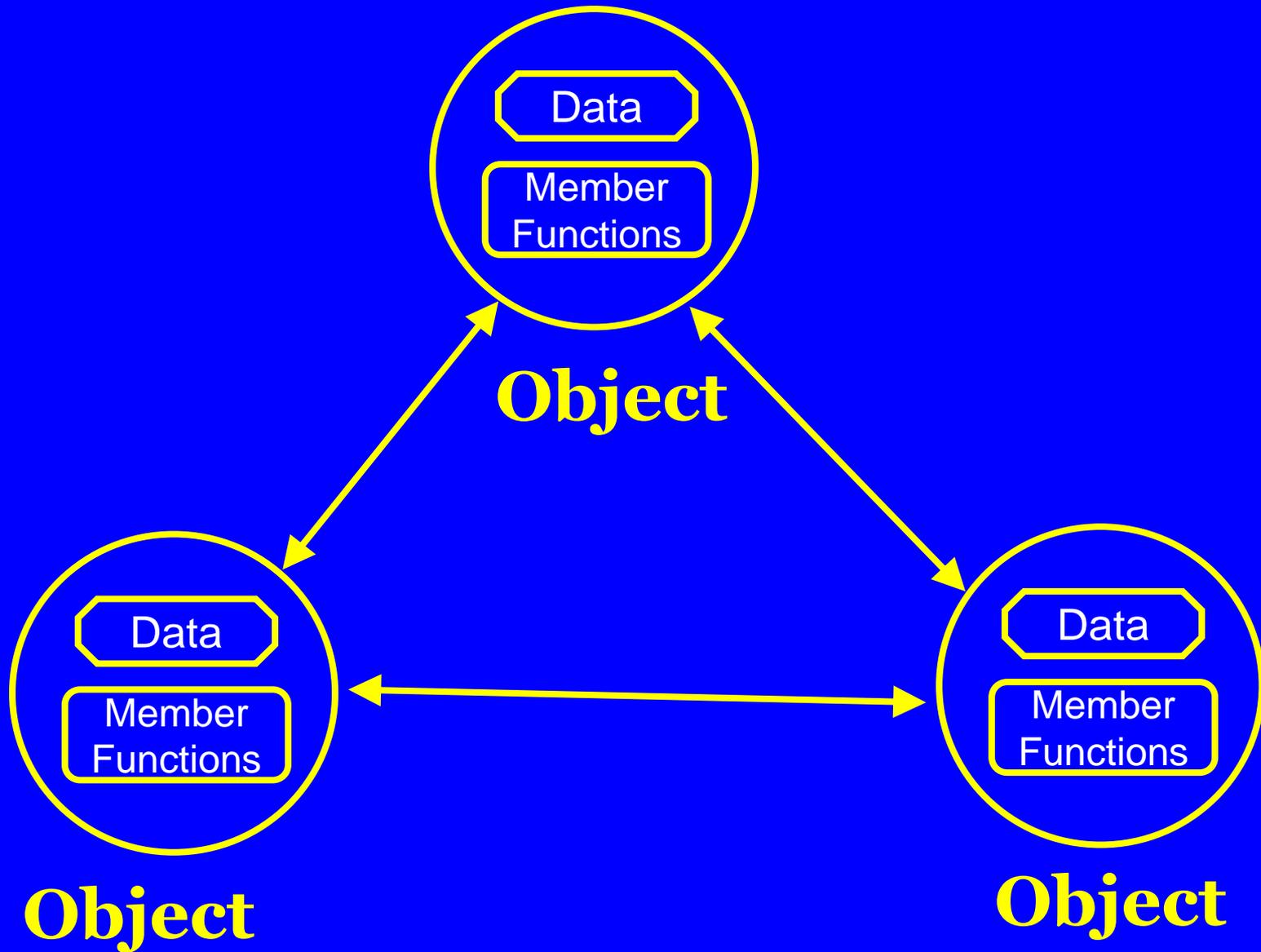
- ✓ Data Abstraction
- ✓ Data Encapsulation
- ✓ Modularity
- ✓ Inheritance
- ✓ Polymorphism

Data Abstraction

Abstraction refers to the act of representing essential features without including the background details or explanations.

What is **encapsulation** ?

The wrapping up of data and operations / functions (that operate on the data) into a single unit (called class) is known as *Encapsulation*.



Modularity

The act of partitioning a program into individual components is called modularity. The justification for partitioning a program is that

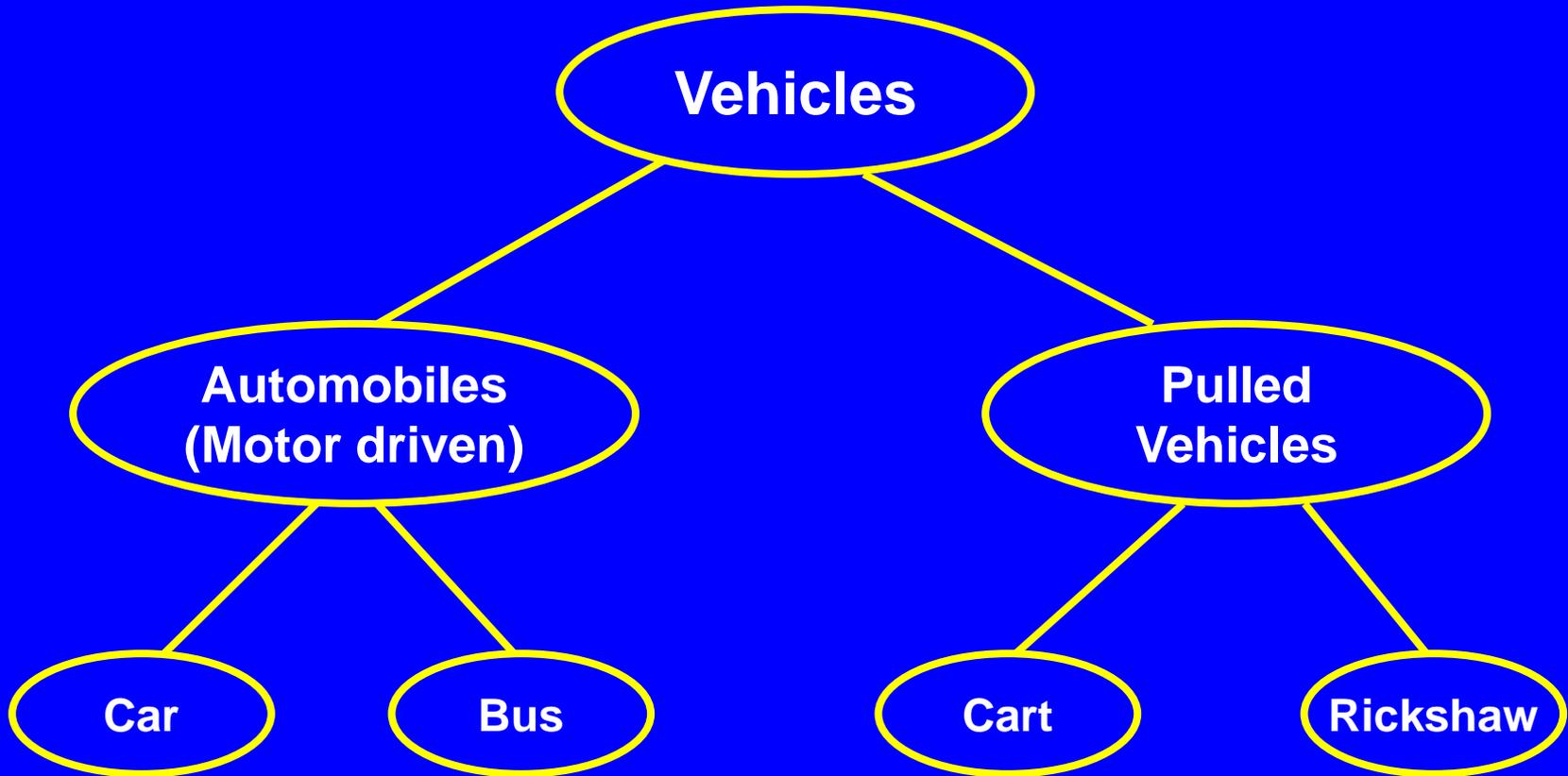
- it reduces its complexity to some degree and
- it creates a number of well-defined, documented boundaries within the program

What is a *Modularity* ?

Modularity is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules.

Inheritance

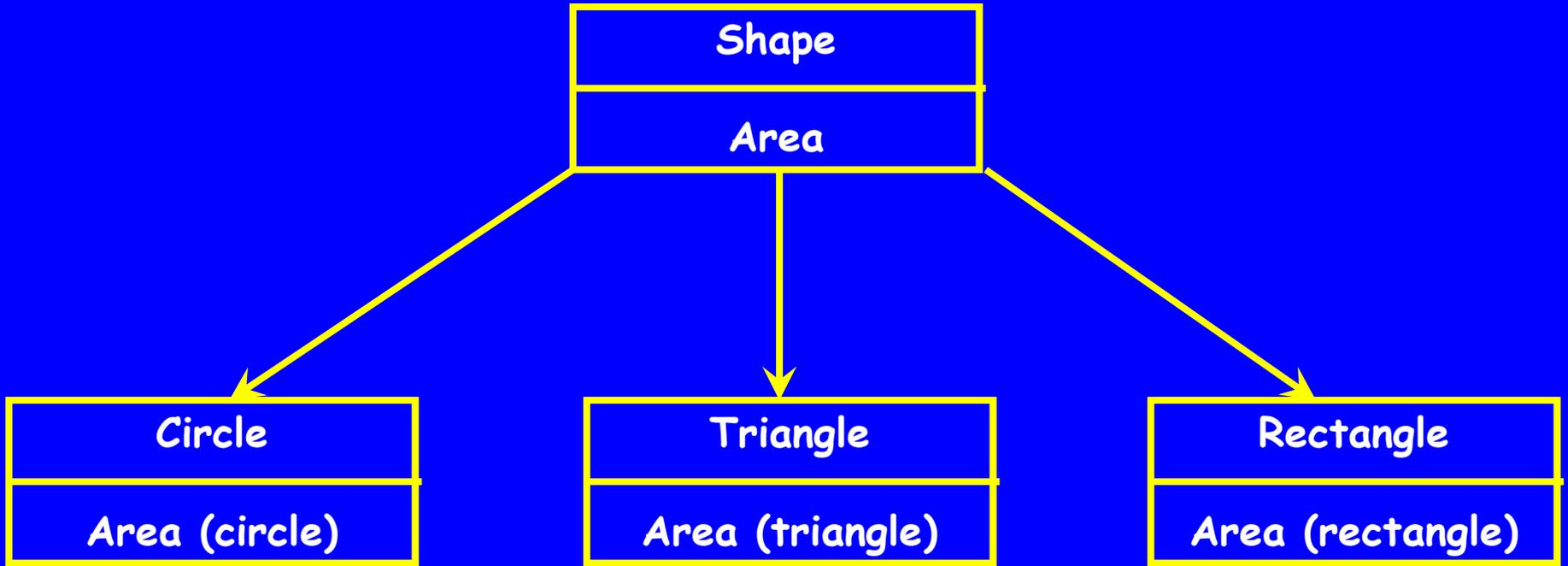
Inheritance is the capability of one class of things to inherit capabilities or properties from another class.



Property Inheritance

Polymorphism

Polymorphism is the ability for a message or data to be processed in more than one form.



Polymorphism

***ADVANTAGES* of OOP**

1. *Re-use of code.*
2. *Ease of comprehension.*
3. *Ease of fabrication and maintenance.*
4. *Easy redesign and extension.*

***DISADVANTAGES* of OOP**

- a) With OOP, classes tend to be overly generalized.
- b) The relations among classes become artificial at times.
- c) The OOP programs' design is tricky.
- d) Also one needs to do proper planning and proper design for OOP programming.
- e) To program with OOP, programmers need proper skills such as design skills, programming skills, thinking in terms of objects etc.