TATA

TATA ELXSI

OBJECT ORIENTED PROGRAMMING USING C++
Module 1

Learning & Development Team

Concepts Of Object Oriented Programming

Objectives

In this section, you will learn :

- Basics Object Oriented Design Concepts
- Benefits of Object Oriented Programming
- Applications of Object Oriented Programming
- Introduction to Class, Objects, Inheritance, Encapsulation & Polymorphism

Structured Programming

- Emphasis on algorithm rather than data.
- Programs are divided into individual procedures that perform discrete tasks.
- Procedures are independent of each other as much as possible.
- Procedures have their own local data and processing logic.
- Support for modular programming.
- A rich set of control structures are available.
- Maintenance of huge project is tedious and costly
- Example : Pascal and C.

History of OOP Languages

- **SIMULA** 1 (1962) and Simula 67 (1967) are the two earliest object-oriented languages. The work on the Simula languages was done by Ole-John Dahl and Kristen Nygaard.

- In the early 1969 the first true O-O programming language: **Smalltalk** was beginning developed at the Learning Research Group at Xerox's company.

- In 1979, **Bjarne Stroustrup** began work on "C with Classes", the predecessor to C++.

  Bjarne Stroustrup at Bell Labs adds features to C to form “C with Classes”

  1983 -- Name C++ first used.

- In 1995, **Java** was developed by James Gosling at Sun Microsystems.

Benefits of OOP

- In object oriented programming, the concept of class which is **a collection of data and methods** that relies on the operation of object.
- The concept of class and object brings the **dynamic-ness** within a code and most importantly make the **code reusable** unlike procedural language.
- Object oriented programming provides data hiding so it is more **secure**.
- The trending OOP languages are C++, Java, Python, PHP, JavaScript, Ruby, Perl, Objective-C, Dart, Swift, Scala.

Applications of OOP

- Real time systems.
- Object Oriented Relational database Management System(OORDBMS).
- Artificial Intelligence and Expert System.
- CAD/CAM
- System Software
- Office Automation System
- Neural Networks
- Parallel Programming

Different Versions of C++

- **C++98** (ISO/IEC 14882:1998) is the first edition.
- **C++03** (ISO/IEC 14882:2003) is the second edition.
- **C++11** is the third edition.
- **C++14** is the fourth edition.
- **C++ 17** edition in 2017.

Difference between struct and class

- In a struct, by default all members **are public** and
- In class , by default all members **are private**.

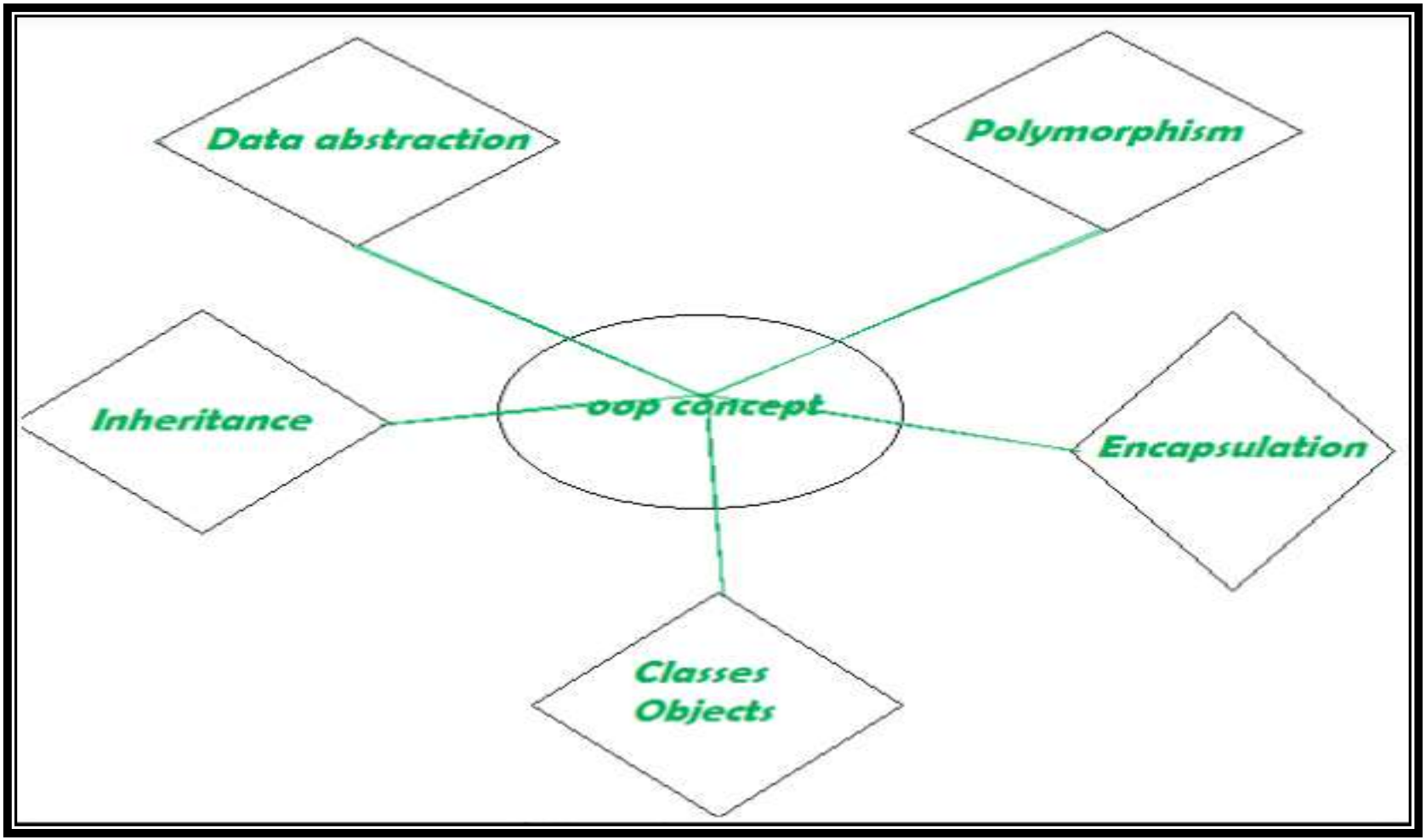
```
struct MyStr
{
   void buildStr(char *s);//public
   void showStr();
  private: // now go private
   char str[255];
} ;
```

```
class MyStr
{
   char str[255];
  public:
   void buildstr (char *s);
   void showstr();
};
```

Object Oriented Programming

- **Object oriented languages** incorporate all the features of object orientation which are commonly known as 5 pillars of OOP.
- They are :
  - Class & Object
  - Abstraction
  - Encapsulation
  - Inheritance and
  - Polymorphism.

Object Oriented Programming
Pictorial representation



Classes

- A class is used to define the nature of an object. (contains both data and functions.)
- Created using the keyword class.

```
class class_name
{
access_specifier_1: member1;
access_specifier_2: member2;
.....
} object_names;
```

- A class declaration is similar to a structure.

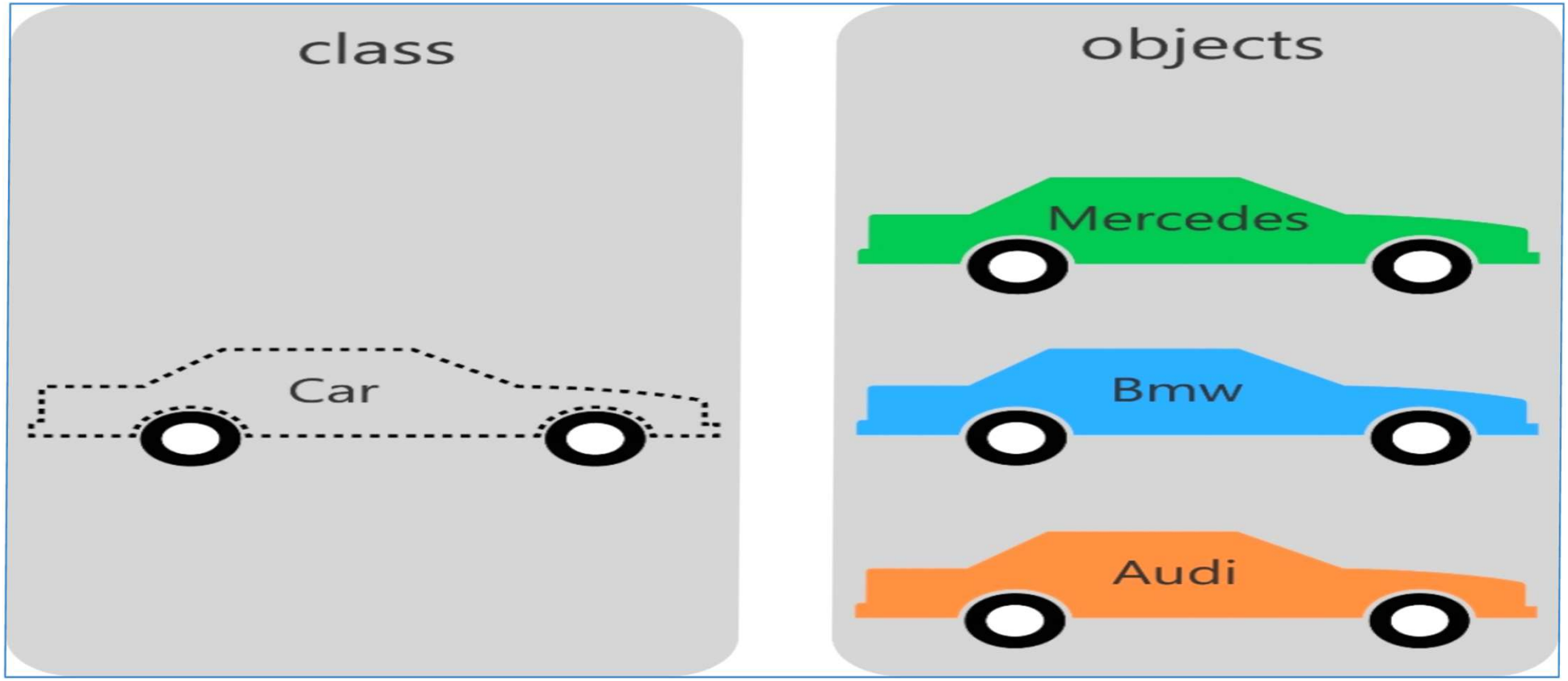
valid identifier for the class

Members could be data or function

Objects

- An object is a physical implementation of a class created in memory by a program.
- An object therefore, represents class instantiation.
- An object is therefore, called as **an instance of a class**.

Classes and Objects



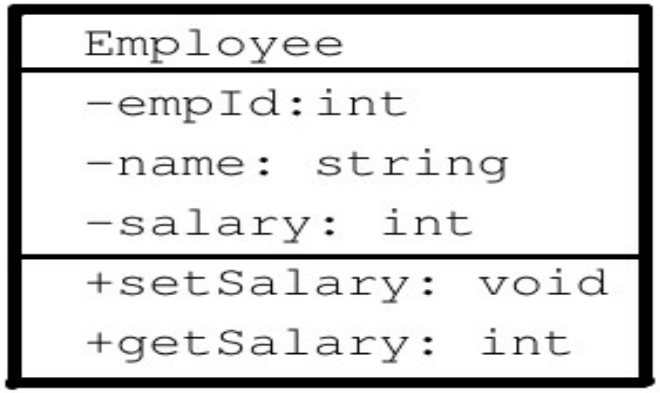
Program Example

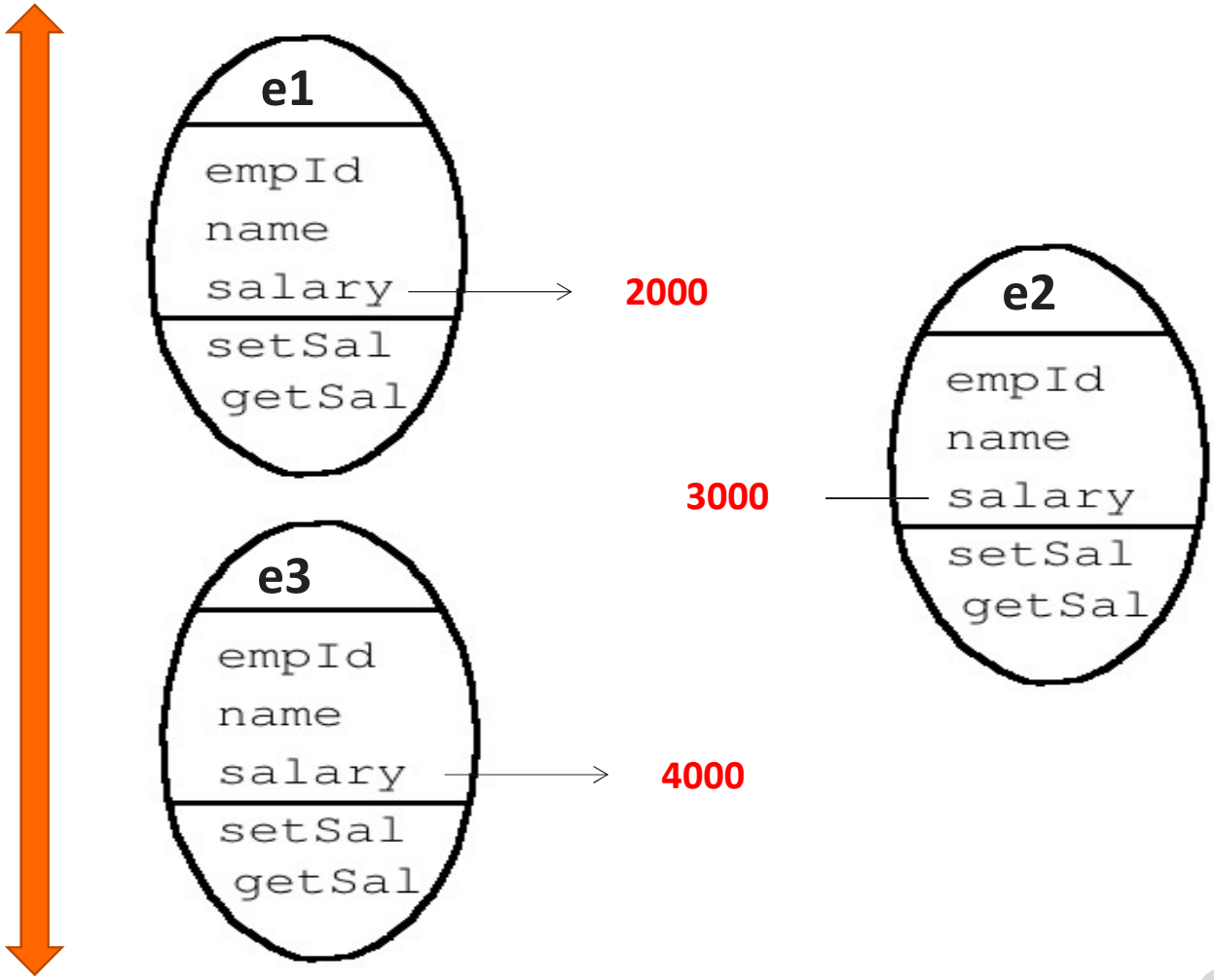
```
class Employee  {
  private:
          int empId;
          string name;
          int salary;
  public:
          void setSalary(int sal){
               salary = sal;
          }
          int getSalary(){
               return salary;
          }
};

main(){
          Employee e1,e2,e3;
          e1.setSalary(2000); e2.setSalary(3000); e3.setSalary(4000);
          e1.getSalary();   e2.getSalary();    e3.getSalary();
}
```

Blue-Print

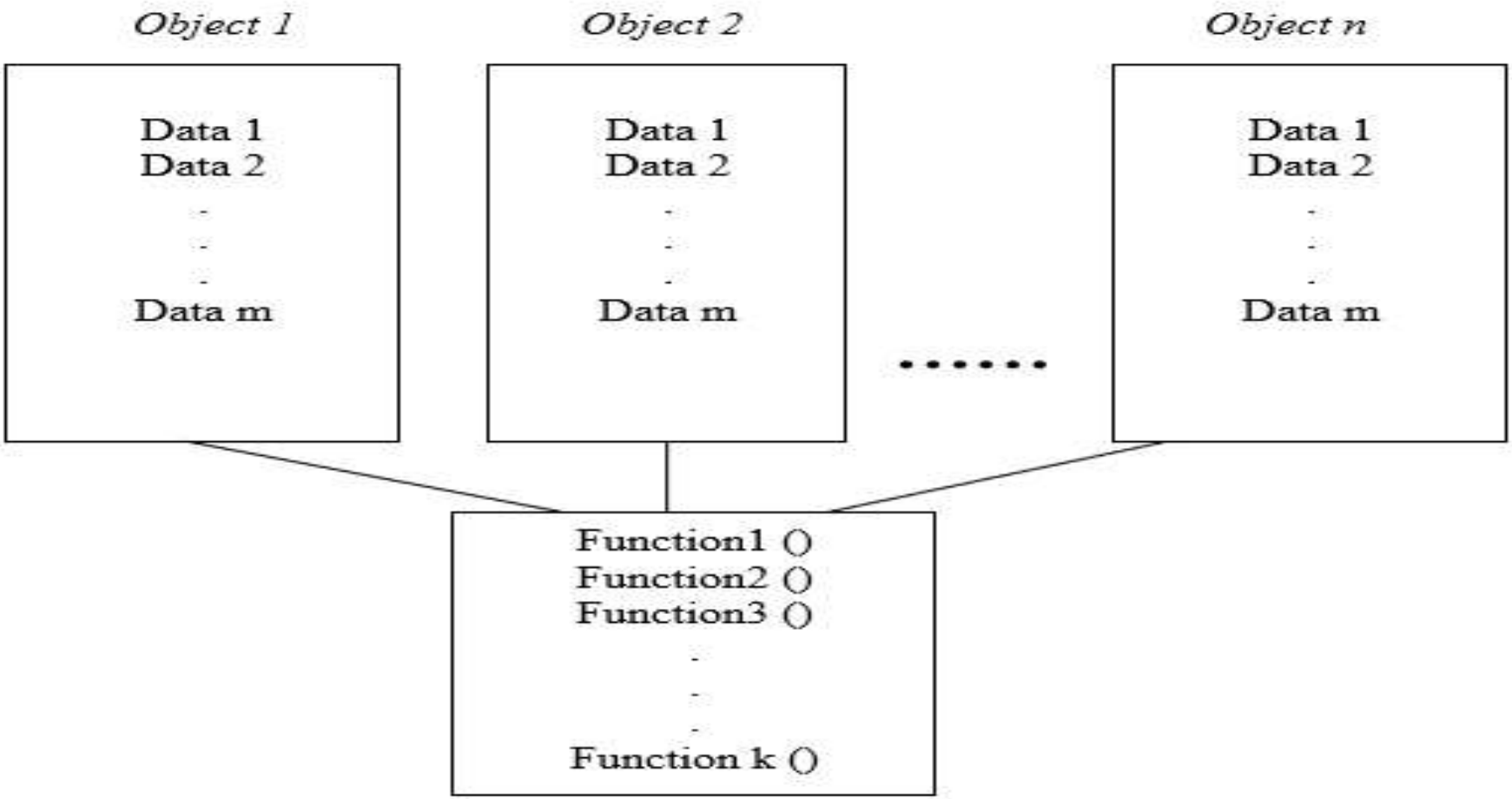
Memory layout of an object

Class Diagram





Classes and Objects: Memory aspects



Class and Object and Memory in C++ Programming

Abstraction [Hide]

- Unable to comprehend complexity of real-world objects while attempting to classify them, we look for what is essential relative to our perspective or understanding. This is Abstraction.

- Abstraction leads to the definition of a well-defined set of public interfaces using which the external world can interact with the object.

- By interacting with these interfaces, the external world can draw out the behaviors of the object, while choosing to ignore the internal implementation of the object.

Encapsulation [Hide & Bind]

- Encapsulation is the process of hiding the implementation-level details of an object.
- Implementation-level details refer to the object's attributes as well as to the implementation of the object's behaviors.
- The internal implementation can only be accessed through the object's public members.
- Keeping attributes and related behaviors together is another way of implementing encapsulation.

Inheritance

- A class that is derived from an already existing class is called as a Derived Class/Subclass.
- The class from which other classes are derived is called the Base class/Superclass
- A subclass not only inherits attributes and behaviours from its superclass, but also adds its own unique attributes and behaviours giving it its unique identity.

Polymorphism

- Polymorphism literally means “anything that is capable of existing in **multiple forms**. (the root words “Poly” and “Morph” are Greek words that stand for “many” and “forms” respectively.

- Polymorphism in OO environments is typically associated with overridden behaviors across subclasses in a class hierarchy, each of which has the same name as that in its super class, but chooses to keep its implementation specific to its own needs.

Are you ready to solve…

1. _____ is called as an instance of a class.

   a. Class b. object c. struct d. inheritance

   Ans: **b. object**

2. _____ emphasizes on reusing the features available.

   a. Abstraction b. Inheritance c. Polymorphism d. Encapsulation.

   Ans: **b. Inheritance**

End of Module - 1

Disclaimer

- Some examples and concepts have been sourced from the below links and are open source material
  - ❖ **http://cppreference.com**
  - ❖ **www.cplusplus.com**

- References:
  - ❖ *C++: The Complete Reference* - 4th Edition by Herbert Schildt, Tata McGraw-Hill publications.
  - ❖ *The C++ Programming Language*- by Bjarne Stroustrup.
  - ❖ *Practical C++ Programming*- by Steve Oualline, O'Reilly publications.

Learning & Development Team

ITPB Road Whitefield

Bangalore 560 048 India
Tel +91 80 2297 9123
Fax +91 80 2841 1474
e-mail info@tataelxsi.com

www.tataelxsi.com