CHAPTER 1

INTRODUCTION TO OBJECT ORIENTED PROGRAMMING.

[3HOURS]

1.1.BASIC OF OBJECT ORIENTED PROGRAMMING:

- OOP is an engineering approach to built software system.
- Object Oriented Programming is a programming paradigm that uses classes and objects to create models based on real world environment.
- Objects are able to pass, receive messages or process information in the form of data . It use the real world concept of object, inheritance and polymorphism.

REASONS FOR USING OOP ARE:

- OOP makes it easy to maintain and modify existing code as new objects are created inheriting characteristics from existing one.
- OOP is the ease of development and ability for other developers to understand the program after development.
- Many programming languages using OOP will destroy or dump unused objects or classes, freeing up system memory. Due to which, System can run the program faster and more effectively.

PROCEDURE ORIENTED PROGRAMMING:

- Procedure Oriented Programming is also referred to as inline programming, takes a more top-down approach to programming.
- POP takes an applications by solving problems from the top of the code down to the bottom.
- Program starts with a problem and then breaks that problem down into smaller sub-problems or sub-procedures.

ISSUES WITH POP:

The main issue in POP is that, if an edit is needed to the program, the developer must edit every line of code that corresponds to the original change in the code.

Example of POP is:

If at the begining of a program, a variable was set equal to the value of 1.If other sub-procedures of the program rely on that variable equaling 1 to function properly, they will also need to be edited.

As more and more changes may be needed to the code, it becomes increasingly difficult to locate and edit all related elements in the program.

COMPARISON BETWEEN OOP AND POP:

OOP	POP
methods	procedures
objects	records
classes	modules
messages	procedure calls
data-fields	procedures

PROCEDURE ORIENTED VERSUS OBJECT ORIENTED PROGRAMMING:

1. In OOP, program is divided into parts called objects. Whereas, in POP, program is divided into small parts called functions.

2. In OOP, importance is given to the data rather than procedure or function. Whereas, in POP, importance is given to function or procedure rather than data.

3. OOP has access specifiers named public, private, protected, etc. Whereas, POP does not have any access specifiers.

4. In OOP, data can not move easily from function to function, it can be kept public or private so we can control the access of data. Whereas, in POP, most function uses global data for sharing that can be accessed freely from function to function in the system.

5. In OOP, data can move and communicate with each other through member function. Whereas, in POP, data can move freely from function to function in the system.

6. With OOP, designs can be reused and recycled throughout the program. Whereas, POP is not able to do this.

7. OOP is bottom-up approach. Whereas, POP is top-down approach.

8. OOP has more abstraction and more flexibility power than that of POP.

9. In OOP, overloading is possible in the form of function overloading and operator overloading . Whereas, in POP, overloading is not possible.

10. OOP provides data hiding so it is more secure. Whereas, POP does not have any proper data hiding process so it is less secure.

11. Example of OOP: C++, JAVA, etc. Whereas, Example of POP: C, FORTRAN, Pascal etc.

ADVANTAGE OF OOP:

- It is reliable and sustainable across different platform.
- OOP is the ease of debugging the code.
- OOP has logical structure. Also, system can be easily upgraded from small to large systems.
- Easy to understand.
- The principle of data hiding helps the programmer to build secure programs.
- Code reusability is much easier than conventional programming languages.

DISADVANTAGES OF OOP:

- In OOP, there is the difficulty of understanding how objects, classes, methods, action etc relate to each other.
- OOP has requirement to have packages and libraries installed for the code to function properly.
- Compiler and runtime overhead is high because OOP requires more time during compilation and dynamic and runtime support it requires more resources and processing time.
- Requires the mastery in software engineering and programming methodology.

CONCEPT OF OOP:

1. Object:

Objects are the entities in an object oriented system through which we perceive the world around us. Object is an instance of class. They may represent a person, a place, a bank account or any item that the program must handle.

Example : Automobiles are objects as they have size, weight, color etc as attributes (i.e. data) and starting, pressing the brake , etc as operation (i.e. functions). Actually, any given automobiles may have its own size, weight, color etc but the operations or tasks are common to all the automobiles.

FIG: MODEL OF GENERAL OBJECT.

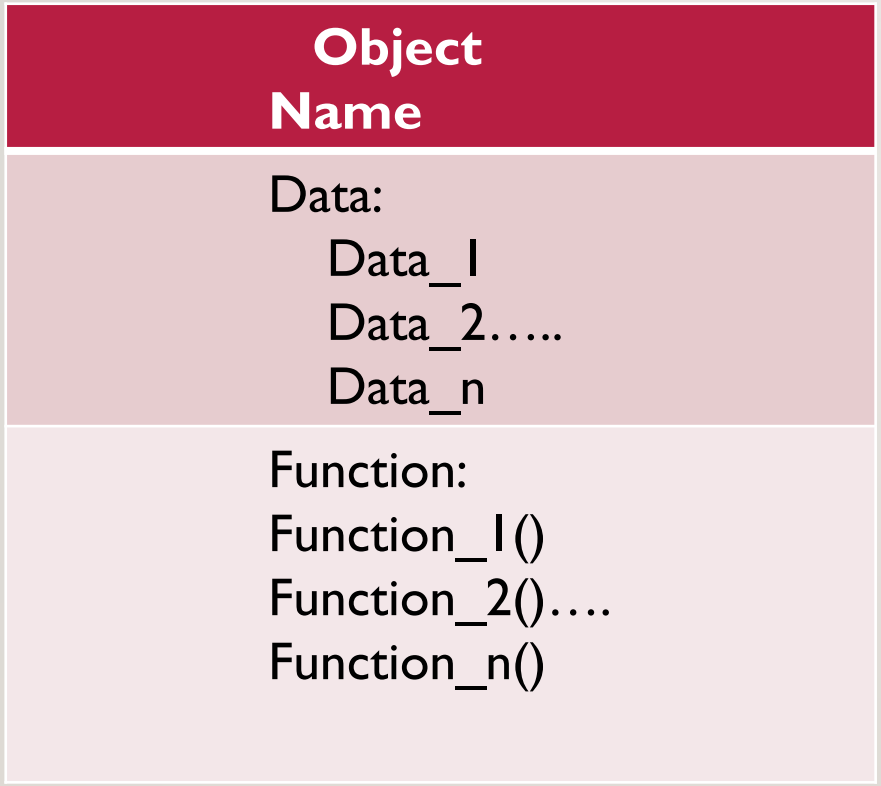


FIG : MODEL OF STUDENT AS OBJECT.

Object: Student
Data: Name Roll No. Marks
Function: Total() Average() Display()

Following are some of the examples of object in different scenario:

1. Physical Object:

- Bus in traffic system.
- Diode in electronic system.
- Leader in political system.

2. Graphical User Interface:

- Menu
- Button
- Toolbar

3. Data Structure:

- Vector
- Stack
- Tree

4. Discipline of human:

- Actor
- Singer
- Teacher

5. Geometrical Shapes:

- Point
- Line
- Circle

6. User Defined Data:

- Distance
- Currency
- Date

2. Class:

A class is a collection of object. Thus, an object is a specific instance of a class. A class is a template(blueprint or formula) definition of methods and variables in a particular kind of object. Also, the class is the user defined data type used to declare the objects. Class is a Structure where we can define variables and methods to utilize by an objects.

Example of class:

Saving Bank Account is a class. //Where account no. and account holder name are variable.

→ A Person (Object) who has Saving Bank Account(Class).

→ So Many People(Object) have Saving Bank Account(Class).

Just an instance: Saving Bank Account(S.P1, S.P2,S.P3). Here, S is class and P1,P2,P3 are objects.

Class : Student
Data Members: Name Roll No. Marks
Function Members: Sort_ name() Tot_ marks() Percentage_ marks()

Object of Student can be:

Name : Ram Thapa

Roll No. : 2

Marks: { 50,60,40,45}

The function Sort_ name() will sort and display list of students on the basis of name in alphabetical order. Similarly, function Tot_ marks() will sum the marks obtained by the Student. The function Percentage_ marks() will calculate the percentage.

CLASS VERSUS OBJECT:

1. **Definition:**

Class is a mechanism of binding data members and associated methods in a single unit. Whereas, Object is an instance of class or variable of a class.

2. **Existence:**

Class has logical existence. Whereas, object has physical existence.

3. **Memory Allocation:** In class, memory is not allocated, when it is created. Whereas, in object, memory is allocated, when it is created.

4. **Declaration/Definition:** In class, definition is created once. Whereas, in object, definition is created many times as you requires.

5. Without class, object does not exist. Whereas, without object, we can use only static function of class.

6. Class generates the objects whereas object gives life to class.

7. Class cannot be manipulated because it is not available in memory (except static class). Whereas, Object can be manipulated.

3. Abstraction:

Abstraction is central principle along with encapsulation and inheritance. Abstraction is a process through which programmer hides all but relevant data about an object in order to reduce complexity and efficiency.

Its main goal is to handle complexity by hiding unnecessary detail from the user. It only provides the interface to use the service that the object provides.

It provides only essential information about the data to the outside world , hiding the background detail or implementation.

Example:

Consider a real life example of a man driving a car. The man only knows that pressing the accelerator will increase the speed of car or applying brakes will stop the car but he does not know about how on pressing accelerator, the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of accelerator, brakes etc in the car. That is, what we call abstraction.

4. Encapsulation:

Encapsulation is the mechanism of combining data and function together into a single unit. It is one of the fundamental concepts in OOP. It describes the idea of bundling data and methods that work on that data within one unit.

Example: a class in JAVA.

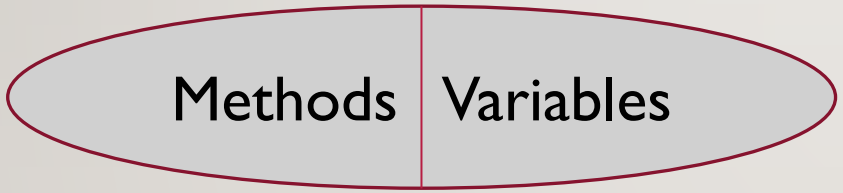
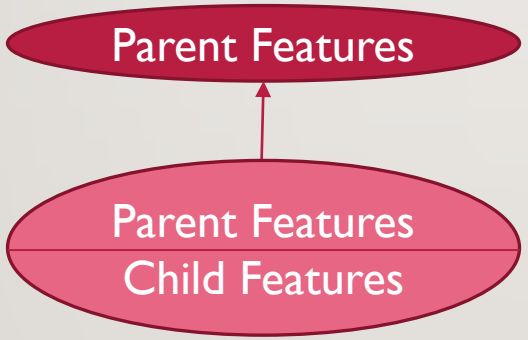


fig: Encapsulation in C++.

5. Inheritance:

It is the process by which objects of one class acquire the properties of another class. It supports the concept of hierarchical classification. In OOL, the concept of inheritance provides idea of reusability.



Parent class or base class or super class

child class or derived class or sub class

Fig: child class inherited from parent class.

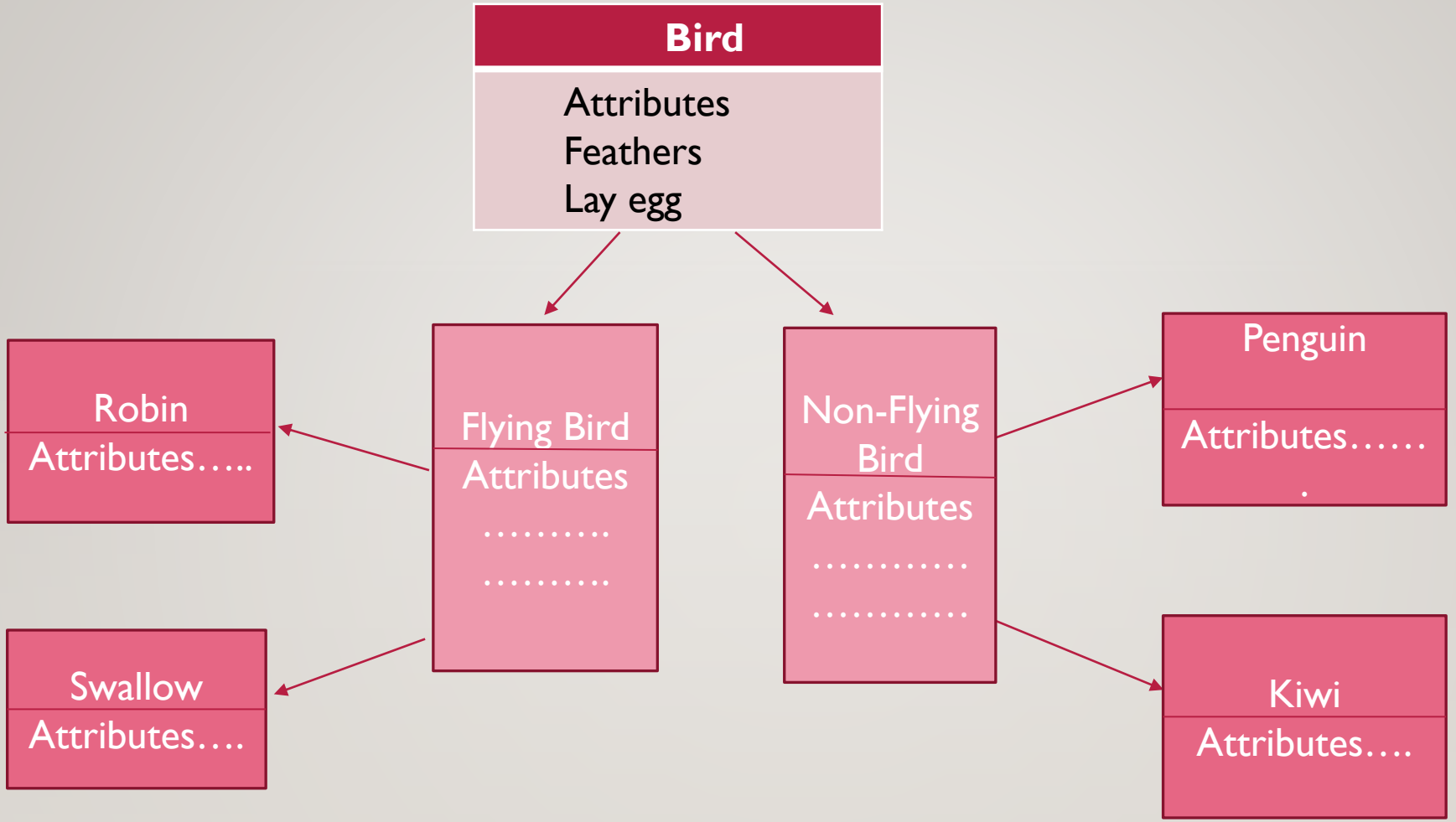


FIG: PROPERTY INHERITANCE.

6. Polymorphism:

Polymorphism means “having many forms” is another important characteristic of OOL. Polymorphism , a Greek term, means the ability to take more than one form.

Operator Overloading ,Function Overloading are example of polymorphism.

A single function name can be used to handle different types of arguments as in figure.

Real Life Example:

A person at a same time can have different characteristic .Like a man at a same time is father, a husband, a employee. So, a same person posses different behavior in different situations. This is called polymorphism.

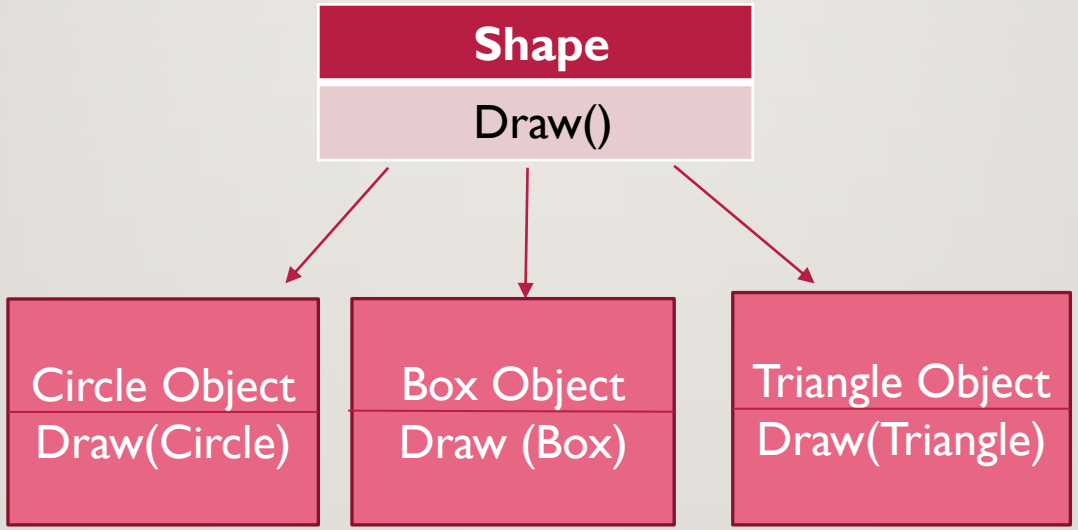


FIG: POLYMORPHISM.

7. Messaging Passing:

It is nothing but sending and receiving of information by the object same as people exchange information. So, this helps in building system that simulate real life.

Following are the basic steps in message passing:

a. Creating classes that defines object and its behavior.
b. Creating objects from class definition.
c. Establishing communication among objects.

In oops, message passing involves specifying the name of objects, the name of the function, and the information to be sent.

E.g. Student . GetMarks (10);

Here, Student is object name.

GetMarks () is message and

10 is information.

8. Dynamic Binding:

Binding refers to the linkage of a procedure call to the code to be executed in response to the call.

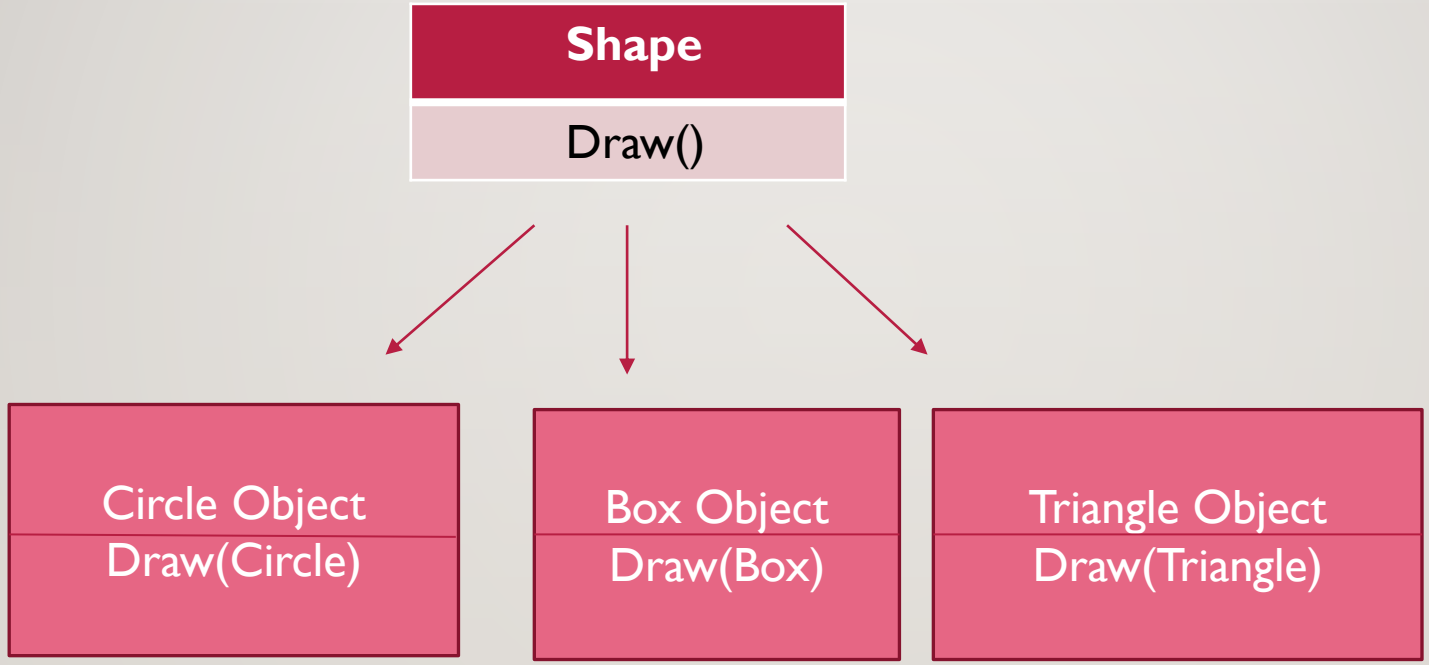
There are two types of binding:

a. Static Binding(or, Early Binding)

b. Dynamic Binding(or, Late Binding)

Dynamic Binding means that the code associated with a given procedure call is not known until the time of the call at run _time. It is associated with polymorphism and inheritance.

E.g. If classes Circle, Box, Triangle are derived from same class shape and all of the function defines a function Draw() , then during the function call Draw() through the pointer variable an appropriate function belonging to that class is involved.



COMPARISON BETWEEN STATIC BINDING AND DYNAMIC BINDING:

	Static Binding	Dynamic Binding
1.	Events occur at compile time are static binding.	Events occurs at run-time are dynamic binding.
2.	All information needed to call a function is known at compile time.	All information need to call a function come to know at run time.
3.	Also known as early binding.	Also known as late binding.
4.	It makes execution of program faster.	It makes program execution slower.
5.	E.g. Overloaded function call, Overloaded operators.	E.g. Virtual function in C++, overridden methods in java.

EXAMPLE OF SOME OBJECT ORIENTED LANGUAGES:

1. **C++:** C++ is a general purpose programming language developed by Bjarne Stroustrup and is an extension of C language. It has imperative, object oriented and generic programming features. It provides facilities for low-level memory manipulation. It is therefore possible to code C++ in a "C style" or "Object Oriented style".

Example:

```
#include<iostream.h>
#include<conio.h>
void main(){
cout<<"Hello World";
getch();
}
```

2. JAVA: JAVA is a general purpose computer programming language that is concurrent, class-oriented , object oriented. It was developed by sun micro-system (now owned by Oracle corporation). JAVA is fast, secure and reliable. From laptops to supercomputers, cell phones to the internet, JAVA is everywhere.

Example: Program of JAVA to print Hello World.

```
public class Hello{

public static void main(String[]args){

System.out.println("Hello World");

}}
```

3. SmallTalk: SmallTalk is an object oriented, dynamically typed, reflective programming language. SmallTalk was created as the language to underpin the "new world" of computing exemplified by "human computer symbiosis".

Example program of SmallTalk:

Transcript show : 'Hello, World!'

4. Eiffel: Eiffel is an object oriented programming language designed by Bertrand Meyer and Eiffel software. It emphasis on the production of robust software. Its syntax is keyword oriented in the ALGOL and PASCAL tradition. Eiffel is strongly statically typed, with automatic memory management.

Example program:

```
class

HELLO_WORLD

create

make
```

(continue……)

```
feature

make

do

Print(“Hello, World!”)

end

end
```

Assignments: Application of OOP, Limitation of POP.