INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

CHAPTER 1

KABIN DEVKOTA

UDITA BISTA

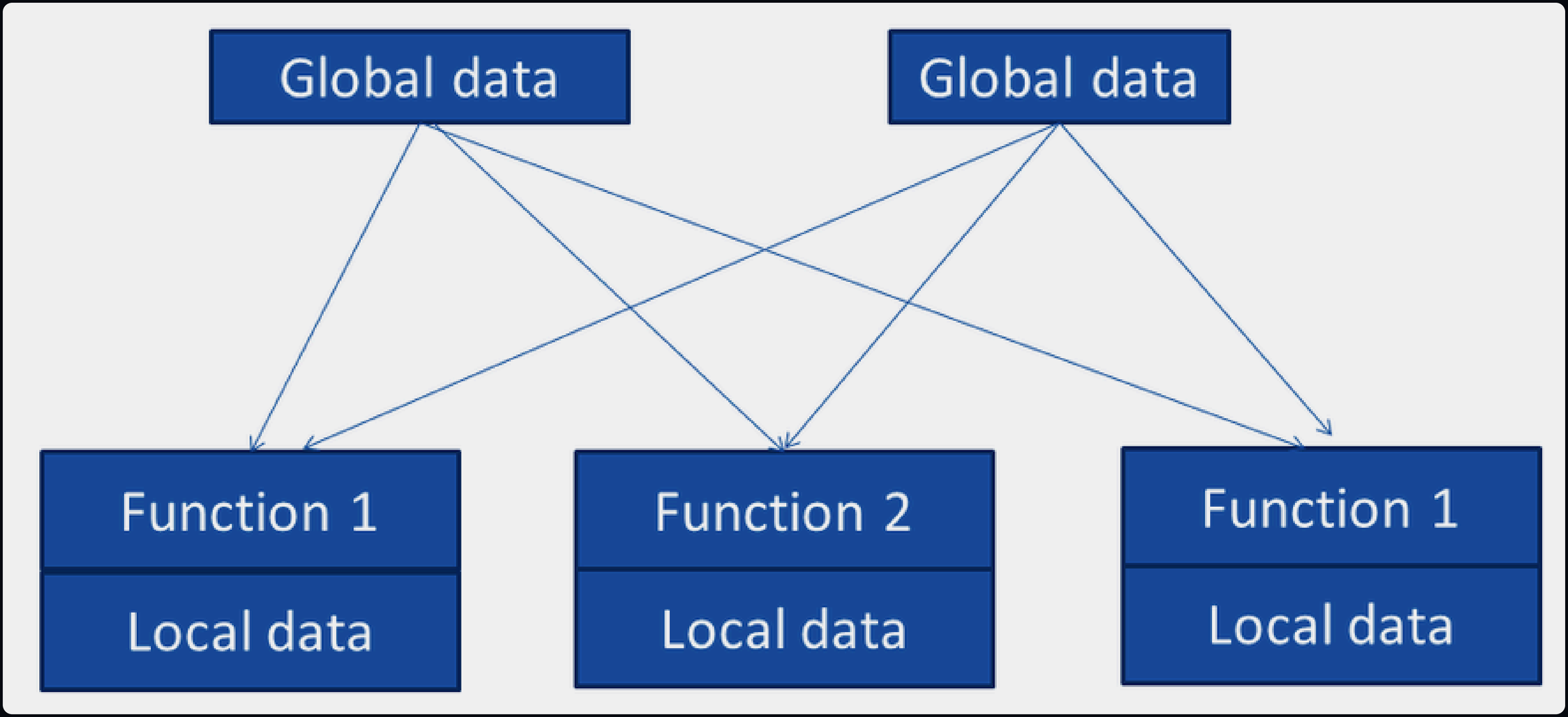
TABLE OF CONTENTS

Our content is
divided into the given topics.
Each part will be described in the slides that follow

ISSUES WITH PROCEDURE ORIENTED PROGRAMMING

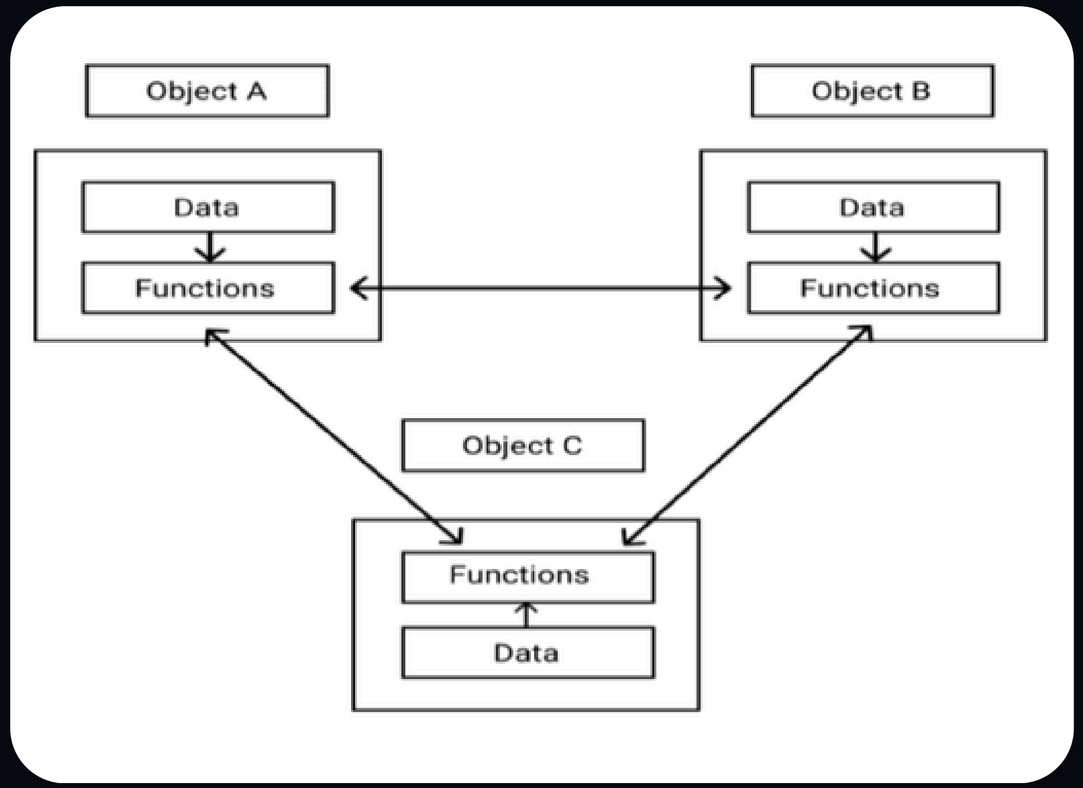
- Emphasis on algorithm (or procedure) rather than data.
- Large programs are divided into smaller programs known as functions. Change in data type being processed needs to propagate to all the functions that use the same data type.
- Procedural languages are difficult to relate with the real world objects.
- As the size of the program increases, readability, maintainability, and bug-free nature of the programs decrease.
- In large programs, It is difficult to identify the belonging of global data.
- The data, which is used in procedural languages are exposed to the whole program. So, there is no security for the data.

Access to data in procedure oriented programming



BASICS OF OBJECT ORIENTED PROGRAMMING

- To overcome the issues caused by procedural programming, OOP was introduced.
- OOP treats data as the critical element in program development.
- In OOP, data associate with the functions that operates on it and protects from any accidental modifications from outside function.
- OOP decomposes the problem into a number of entities called objects and then builds data and functions around these objects.

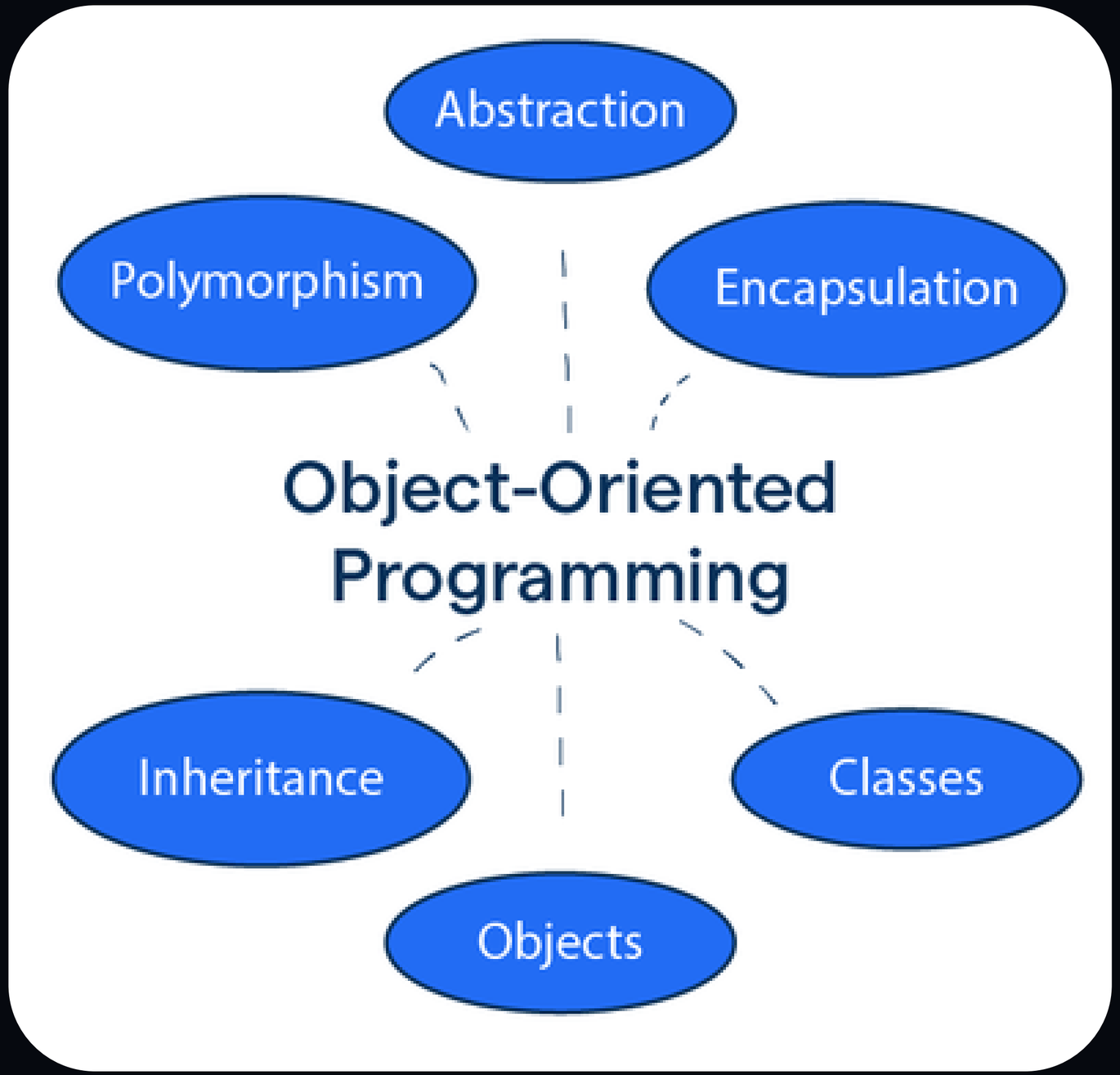


FEATURES OF OBJECT ORIENTED PROGRAMMING

1. Emphasis is on data rather than procedure
2. Programs are divided into objects.
3. Functions that operate on the data of an object are tied together in the data structure.
4. An object is a complete entity, it has all the data and associated functions within it. So, whenever something is to be changed for an object, only its class gets affected.
5. Data is hidden and cannot be accessed by external functions.
6. Objects communicate with one another through the process called message passing.
7. Follows bottom-up approach in program design.

Procedure Oriented Programming	VS	Object Oriented Programming
1) Procedural Programming divides the program into small programs and refers to them as functions.		1) Object Programming divides the program into small programs and refers to them as objects.
2) Focuses on process/logical structure and then data required for that process.		2) Object oriented Programming is designed which focuses on data.
3) Procedure Oriented Programming follows a top-down approach.		3) Object oriented Programming follows a bottom-up approach.
4) Data and functions don't tie with each other.		4) Data and functions are tied together
5) Procedure oriented programming is less secure as there is no way of datahiding.		5) Object oriented programming is more secure as it has a data hiding feature.
6) Procedure Oriented programming can solve moderately complex programs.		6) Object oriented programming can solve any complex program.
7) Provides less reusability and more function dependency.		7) Provide less function dependency and more reliability.
8) Data moves free around the system from one function to another		8) Data is hidden and cannot be accessed by external functions.
Eg: C, ALGOL, Fortran, Pascal, etc.		Eg: C++, python, Java, C#(C Sharp), etc.

CONCEPTS OF OBJECT ORIENTED PROGRAMMING



CONCEPTS OF OOP

I. Class

- It is the combination of data members and member functions put together into a single unit.
- It is a group of objects that share common properties and relationships.
- Objects are the variables of the type class. Once a class has been defined, we can create any number of objects belonging to that class.

II. Object

- Object is an instance of a class that encapsulates data and functionality pertaining to that data.
- They have some characteristic and behaviors.
- They may represent a person, a place, a bank account, a table of data or any item that the program has to handle.
- In OOP, the characteristic of an object means the data and its behavior is represented by its associated functions

CONCEPTS OF OOP

III. Abstraction

- Abstraction is the process of hiding implementation details while providing a simplified interface for interacting with the object.
- This helps to reduce complexity and improve maintainability.
- To understand abstraction, let us consider an example of switch board. We know only to press switches according to our requirements. This is, we know which switch to press for which light or machine but we don't know what is happening inside the switchboard. This is abstraction, we know essential features but we don't know about the background details.

CONCEPTS OF OOP

IV. Encapsulation

- Encapsulation is known as the mechanism of wrapping up of data and function in a single unit.
- The only way to access data is through functions that are combined along with data. These functions are called member functions in C++.
- This insulation of data from direct access by the program is called data hiding or information hiding.

CONCEPTS OF OOP

V. Inheritance

- Inheritance is the ability of a class to inherit the properties and methods of another class.
- The new class is known as derived class and the class from which it is derived is known as base class.
- In OOP, the concept of inheritance provides the idea of reusability. It allows the addition of additional features to an existing class without modifying it.
- Inheritance shows the transitive nature i.e. if B class is derived from class A and class C is derived from B then C inherits the properties of class A.

CONCEPTS OF OOP

VI. Polymorphism

- Polymorphism means, “having different forms”.
- Polymorphism is the concept that supports the capabilities of an object of a class to behave differently in response to a message or action. An operation may behave differently in different instance.
- Example of polymorphism in C++ is operator overloading, function overloading and another type of polymorphism is achieved during runtime is known as late binding.

ADVANTAGES OF OBJECT ORIENTED PROGRAMMING

1. High code reusability through inheritance, templates.
2. Real world representation.
3. Data hiding through abstraction so that data is secure.
4. Dividing program into number of object makes software development easy.
5. Maintenance and upgrading of software easy.
6. Improve reliability and flexibility.
7. Existing class can serve as library for further enhancement.
8. Message passing makes the interface easy.

DISADVANTAGES OF OBJECT ORIENTED PROGRAMMING

1. If the message passing between many object in an application, it is difficult to trace the error.
2. Compile time and execution time is high, as it requires more time for dynamic memory allocation and runtime polymorphism.
3. The programmer should have a proper planning before designing a program using OOP approach.
4. Require full knowledge of OOP before using it properly.

APPLICATIONS OF OBJECT ORIENTED PROGRAMMING

1. Client-Server System
2. Object Oriented Database
3. Real Time Systems Design
4. Simulation and Modeling System
5. Hypertext, Hypermedia
6. Neural Networking and Parallel Programming
7. Decision Support and Office Automation Systems
8. CIM/CAD/CAM Systems
9. Al and Expert Systems

THANK YOU