

# **FAST**

# National University of Computer and Emerging Sciences Peshawar

OOP Lab # 8.1

## C++ Object Oriented Programming

**Computer Instructor:** Muhammad Abdullah Orakzai

DEPARTMENT OF COMPUTER SCIENCE



Programming



الذى علم بالقلم- علم الانسان ما لم يعلم-



# Contents

- 1) Object Oriented Programming paradigms
- 2) POP vs OOP
- 3) OOP Properties
- 4) Classes And Objects
- 5) Member Data and Member Functions
- 6) Data hiding
- 7) Encapsulation
- 8) Access Specifiers
- 9) Inheritance
- 10) Polymorphism
- 11) Abstraction



# OOP-->Object Oriented Programming

OOP is methodology or paradigm (نمونه) to design a program using class and object.

OOP is paradigm that provides many concepts such as:

- Classes and objects
- Encapsulation (binding code and its data) etc.
- Inheritance
- Polymorphism
- Abstraction
- Overloading

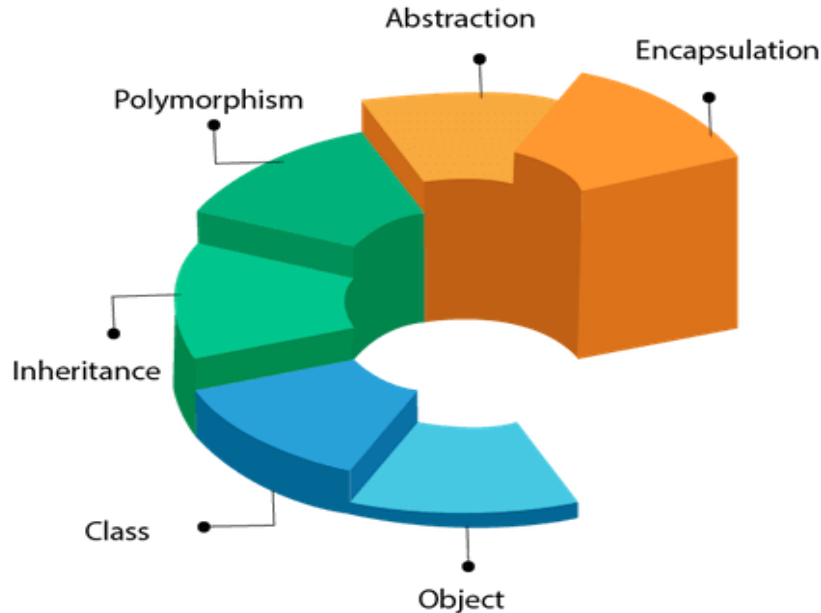


# OOP-->Object Oriented Programming...

- OOP is used to reduce complexity.
- To divide large and complex program into chunks and modules.

# Object Oriented Programming (OOP)...

OOPs (Object-Oriented Programming System)



# Why OOP ?



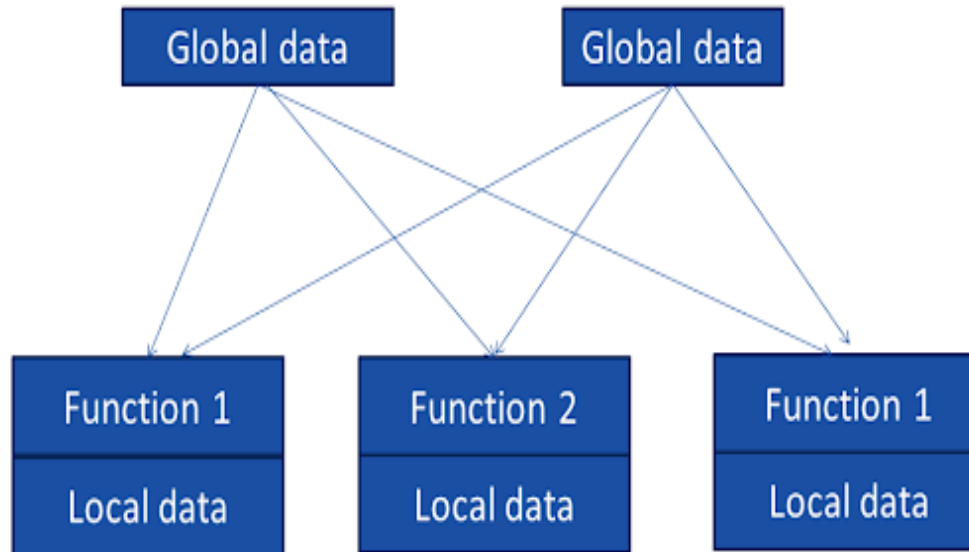
Real world implementation.



# Procedure Oriented Programming (POP)

- Procedural Oriented Programming is one of the programming methods where the main focus is on functions or procedures required for computation, instead of data.
- The program is divided into functions, and the task is done sequentially. These functions share the global data or variables, and there is an exchange of data among those functions.

# Procedure Oriented Programming (POP)...





# Procedure Oriented Programming (POP)...

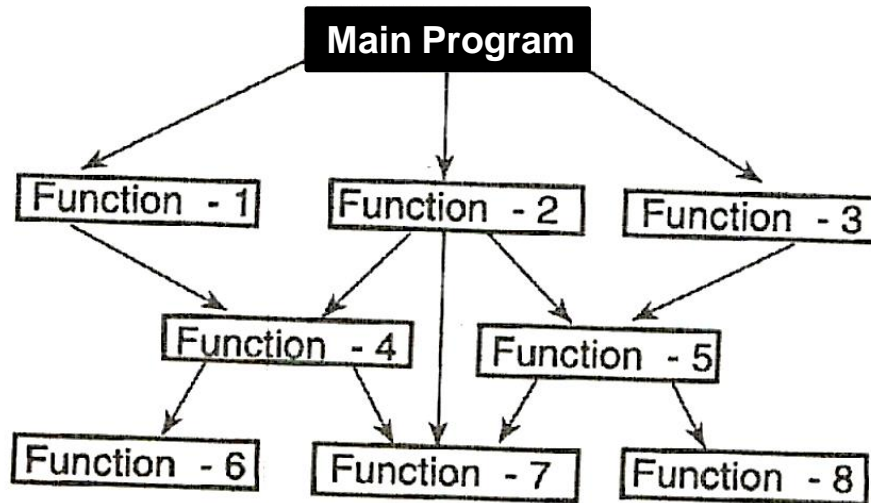


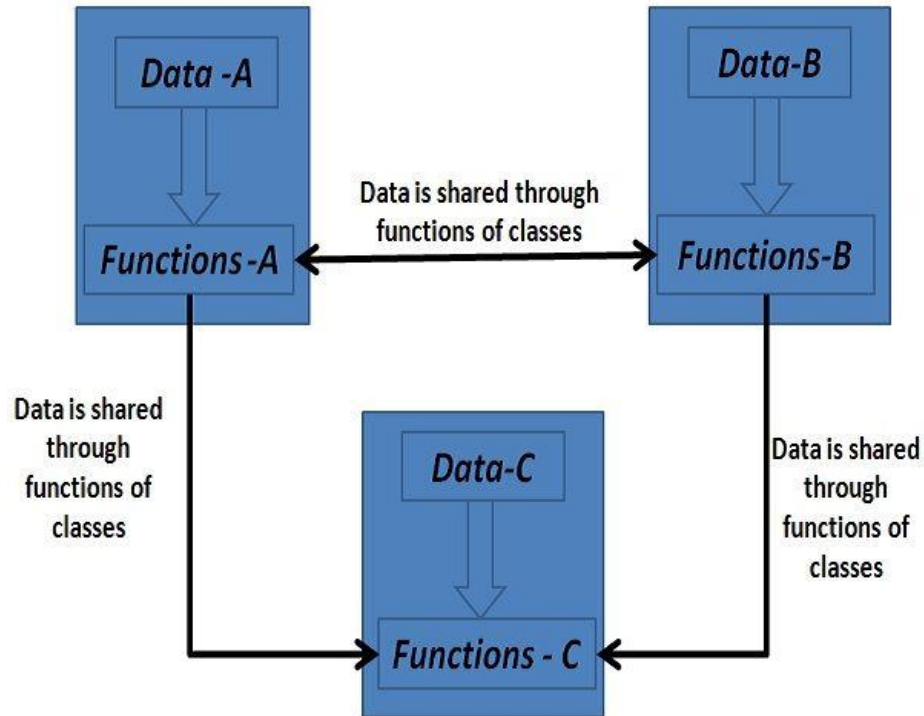
FIG 2.1 Typical structure of procedure-oriented programs



# Object Oriented Programming (OOP)

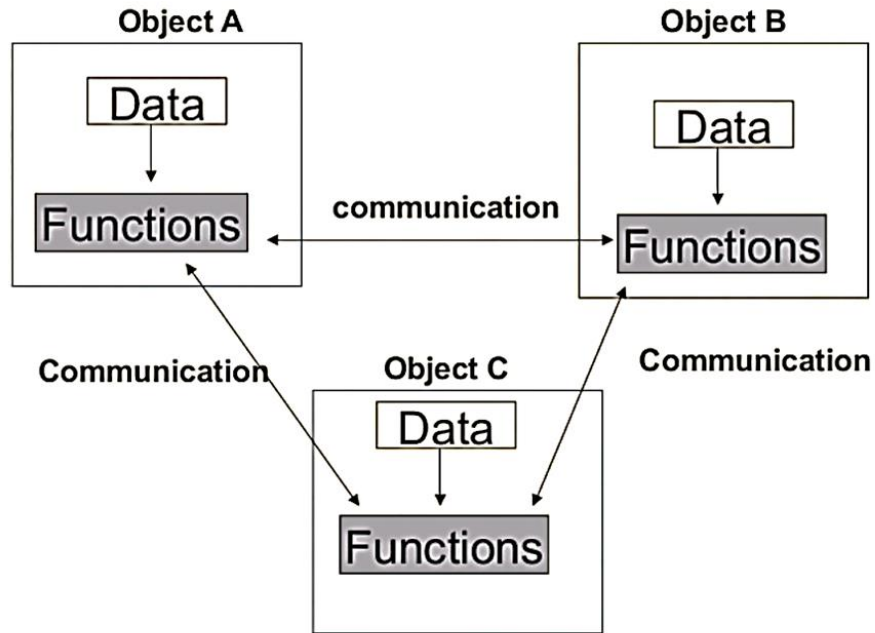
- Object-Oriented Programming is one of the high-level programming languages in which a program is divided into objects.
- Using objects, the programmer can model real-world scenarios. An object is an instance of a class and has state and behavior.
- The state is the attributes, or data, whereas Behavior is called a method.

# Object Oriented Programming (OOP)...



Data flow in object-oriented programming

# Object Oriented Programming (OOP)...



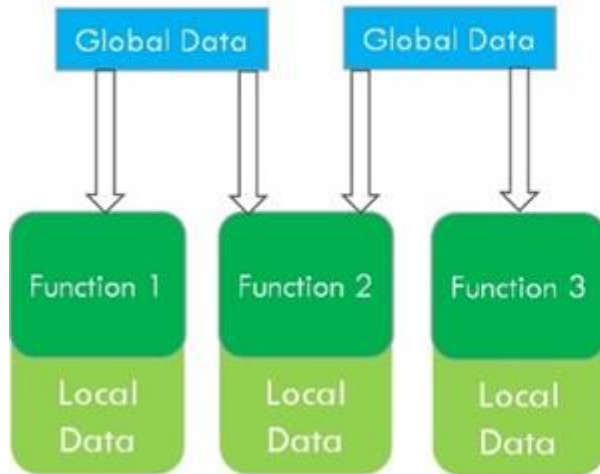


# OOP Vs POP: Comparison Table

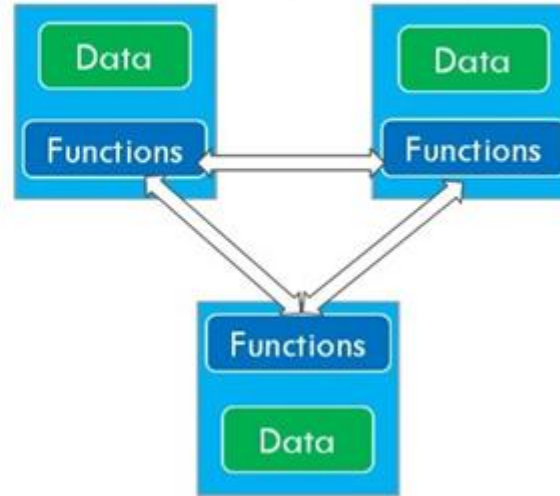
The key difference between OOP and POP is that an OOP divides a program into smaller objects, whereas POP divides a program into smaller procedures or functions to arrive at the results of the problem.

# OOP Vs POP: Comparison Table...

**Procedural Oriented Programming**



**Object Oriented Programming**





# Limitations of POP

- Emphasize is on algorithm or procedure
- Not suitable for modeling a real world problems
- No security and integrity to the data.
- Data cannot be hidden.
- Follows top to down program design.
- Can't reuse the existing code.
- Data will be shared by many functions.
- Difficult to write and understand.



# Advantages/Features of OOP

- Emphasize is on data rather than the procedure.
- Both data and functions are combined into a single unit.
- Data can't be accessed directly.
- Higher productivity.
- Provides multiple instances of an object.
- Provides security to the data.
- Easy to write an understand a program.
- Data is hidden and cannot be accessed by external functions.
- Programs are divided into elements known as objects.





# Advantages/Features of OOP...

- Object may communicate with each other through functions.
- New data and functions can be easily added.
- We can eliminate the redundant code.
- Time will be saved.
- Data is critical element.
- Data can not be freely accessed by external functions.
- OOP permits reusability of existing code.
- We can easily upgrade from small to large systems.



# Advantages/Features of OOP...

- We can define user defined data types.
- Objects are to classes and variables are to data type.



# Class

- Class is blue print or map for object.
- Class is the logical construct of object.
- Class is the description of object.
- Class is a template which contains behaviour (member functions) and attributes/properties (data/variables) of object.
- Means data members and member functions are defined within a class.
- Class is user defined data type because user defined it (non primitive data type).

# Class...



- **Attribute:** Properties abject has.
- **Methods:** Actions that an object can perform.



# Object

- An entity that has state and behaviour.
- An actual existence of a class is called object.
- Object encapsulates data and behaviour.
- When a class template is implemented in real world then it becomes an object.
- Object is the instance of the class.
- Class is the template or blue print from which objects are created, so object is the **instance (result) of the class**.
- The space reserved in memory for class is called an object. Instance of a class. (Instance---→ Single occurrence).



# Object...

- Object is used to perform responsibility of communication between different classes.
- Any entity that has state and behavior is known as an object. For example, a chair, pen, table, keyboard, bike, etc. It can be physical or logical.
- **Example:** A dog is an object because it has states like color, name, breed, etc. as well as behaviors like wagging the tail, barking, eating, etc.

# Object...





# Class VS Object

- **Class:** No data ---> No memory is reserved for class.
- **Object:** Having Data ---> Memory is allocated for class when object of that class is created.





# Member data and member functions

- **Member Data:** The data or the attributes defined within a class is called member data.
- Are variables of any primitive data types.
- It can be any type of array.
- **Member Function:** The functions that are used to work on the data items are called member functions.
- Member functions are used to process and access data members of an object.
- Member functions are functions that are included within the class.

# Encapsulation

- Data and behaviour (function) are tightly coupled inside an object.
- Combine data (variables) and functions in a single container.
- The combining of both data and functions into a single unit.
- To combine code functions and data (variables) in a single box or wrapper.
- In C++ class is the example of encapsulation.
- **Capsule:** It is wrapped with different medicines.



Capsule

Encapsulation in C++





# Data Hiding

- ❖ Means you cannot access data.
- ❖ Making data to be accessed from within the class.



# Access Specifier

- ❖ It specifies that member of a class is accessible outside or not.
- ❖ It may be public, protected or private or default.



# Inheritance

- The mechanism in which one class acquires all the properties and behavior of another class is called inheritance.
- Including features of one class into another class is called inheritance.
- When one object acquires all the properties and behaviors of a parent object, it is known as inheritance.
- It provides code reusability.

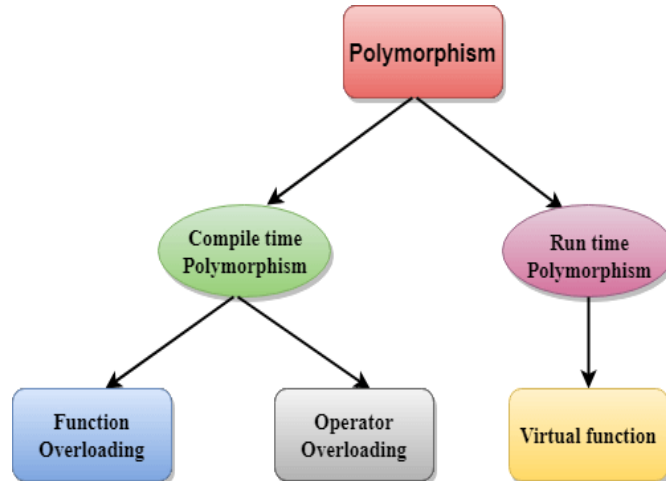


# Polymorphism

- When one task is performed by different ways.
- Poly means many and morphs means shapes.
- **E.g.** to convince customer differently, to draw something e.g. shape or rectangle.
- Another example can be to sound/speak something i.e. cat speaks meaw, dog barks woof etc.
- Let's consider a real-life example of polymorphism. A lady behaves like a teacher in a classroom, mother or daughter in a home and customer in a market. Here, a single person is behaving differently according to the situations.

# Polymorphism

- **Note:** In C++ we use function overloading, operator overloading and virtual functions to achieve polymorphism.
- It may be compile time polymorphism and run time polymorphism.



# Polymorphism...







# Abstraction

- Hiding internal details and showing functionality is known as abstraction.
- **For example** phone call, we do not know the internal processing.
- Data abstraction is one of the most essential and important features of object-oriented programming in C++.
- Abstraction means displaying only essential information and hiding the details.
- Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.



# Abstraction

- Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of the 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.
- **Note:** We can implement Abstraction in C++ using abstract classes and header files.



# Overloading

- Operator overloading is a compile-time polymorphism in which the operator is overloaded to provide the special meaning to the user-defined data type.
- Operator overloading is used to overload or redefines most of the operators available in C++. It is used to perform the operation on the user-defined data type.
- For example, C++ provides the ability to add the variables of the user-defined data type that is applied to the built-in data types.



# Instance variable in C++

- A variable that is created inside a class but outside a method is called instance variable.
- Instance variable does not get memory at compile time.
- It gets memory at runtime when an object (instance) is created. That is why it called instance variable.
- For Examples:

```
int accno; //data member (also instance variable)
```

```
string name; //data member(also instance variable)
```



# static variables or fields

- A field or variable which is declared as static is called static field or static variable.
- Unlike instance field which gets memory each time whenever you create object, there is only one copy of static field created in the memory.
- Static data members is shared by all objects of a class.

`int accno; //data member (also instance variable)`

`string name; //data member(also instance variable)`

`static float rateOfInterest; //static variable`



# References

- <https://beginnersbook.com/2017/08/cpp-data-types/>
- [http://www.cplusplus.com/doc/tutorial/basic\\_io/](http://www.cplusplus.com/doc/tutorial/basic_io/)
- <https://www.w3schools.com/cpp/default.asp>
- <https://www.javatpoint.com/cpp-tutorial>
- <https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/?ref=lbp>
- <https://www.upgrad.com/blog/>

# THANK YOU

