Chapter-2

Concepts of Object Oriented Programming

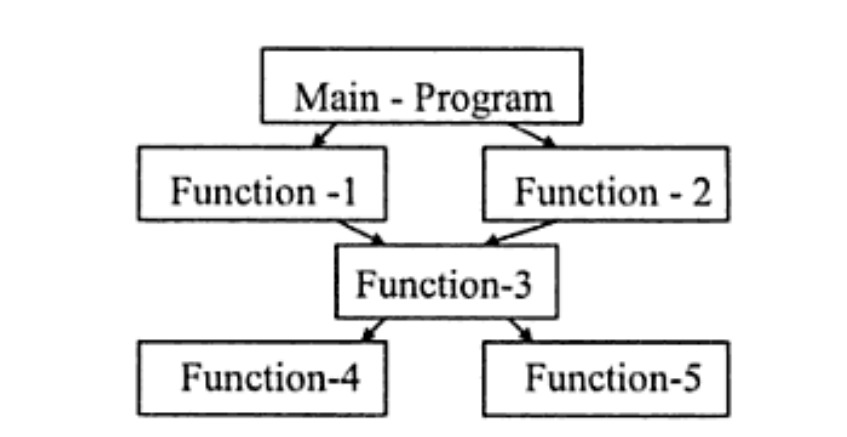
Prepared By
Siraj.F.M
HSST Computer Science
AKGS GHSS Peralassery

Programming Paradigm

- It denotes the way in which a program is organised.
- If it is very small there is no need to follow any organising principle.
- Some paradigm gives more importance to procedure whereas others give importance to data.
- The various approaches in programming are modular,top-down,bottom–up and structured programming.
- C++ implements two types of paradigms,procedural and object oriented paradigm.

Procedure oriented programming paradigm

- Procedure oriented programming consists of a set of instructions and organizes these instructions into functions.
- Programming using high-level languages such C,COBOL is known as procedure-oriented programming.



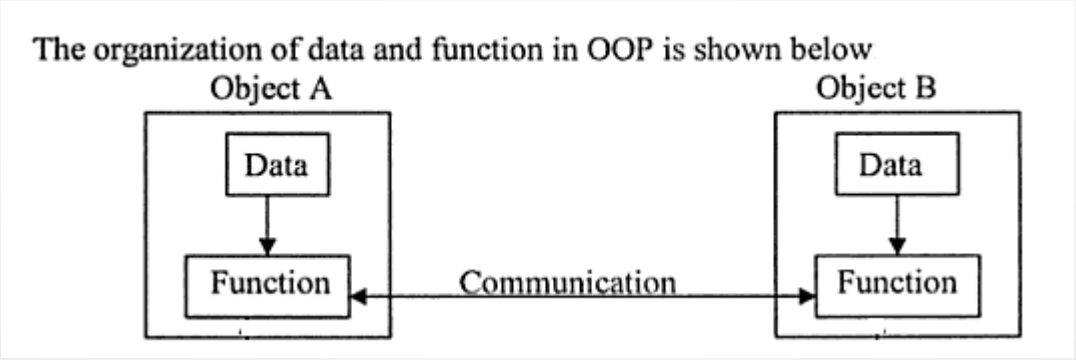
- Here when the program becomes larger and complex the list of instructions is divided and grouped into functions. Here functions clearly define the purpose. To reduce the complexity the functions associated with a common task are grouped into modules.

Limitations of Procedure oriented programming

- **1)Data is undervalued**
- Procedural programming gives importance on doing things. Data is given less importance i.e, any function can access and change data.
- **2)Adding new data element needs modification to functions**
- As functions access global data ,data cannot be changed without modifying functions that access data.
- **3)Difficult to create new data types**
- The ability of a programming language to create new data types is called extensibility.Extensibility helps to reduce program complexity. Procedural languages are not extensible.
- **4)Provides Poor real world modelling**
- In procedural programming data and functions are not considered as a single unit. So it is independent of each other. So neither data nor function cannot model real world objects effectively.

Object Oriented Programming Paradigm

- It eliminates the problems in the procedural paradigm.
- Object Oriented Programming binds data and function into a single unit called Object.
- In OOP program is divided into objects.
- Objects communicate with each other by using functions.

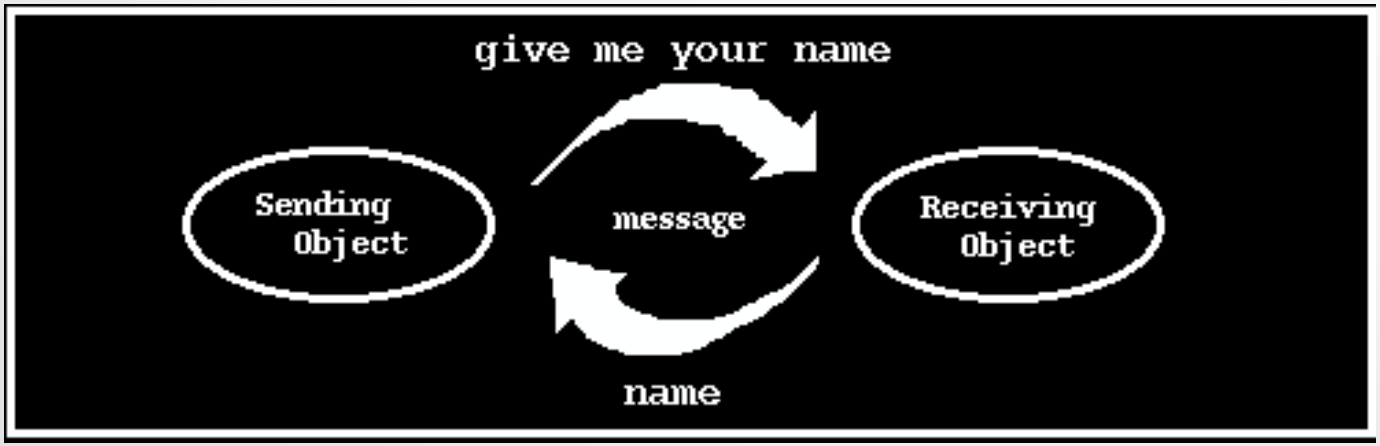


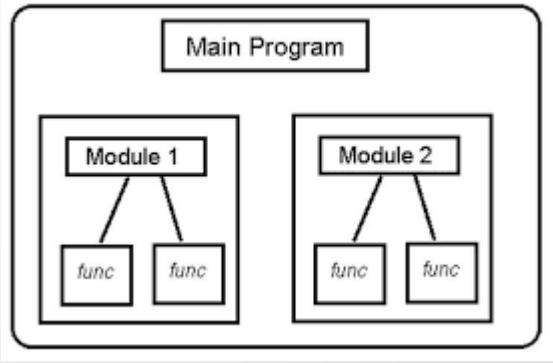
Basic concepts of OOP

- **1)Objects:**-Objects are real world objects such as a person,student,book car,TV ...etc.
- Objects are a combination of data and functions.
- An object has a unique identity,state and behaviour.
- Objects interact with each other by sending messages.
- **2)Classes:**-A class is a **prototype/blue** print that defines data and functions common to all objects of a particular type.
- Classes are user defined data type which contain both data and function.
- A class is a collection of objects.
- An object is an instance of a class. Objects communicate with each other by message passing.

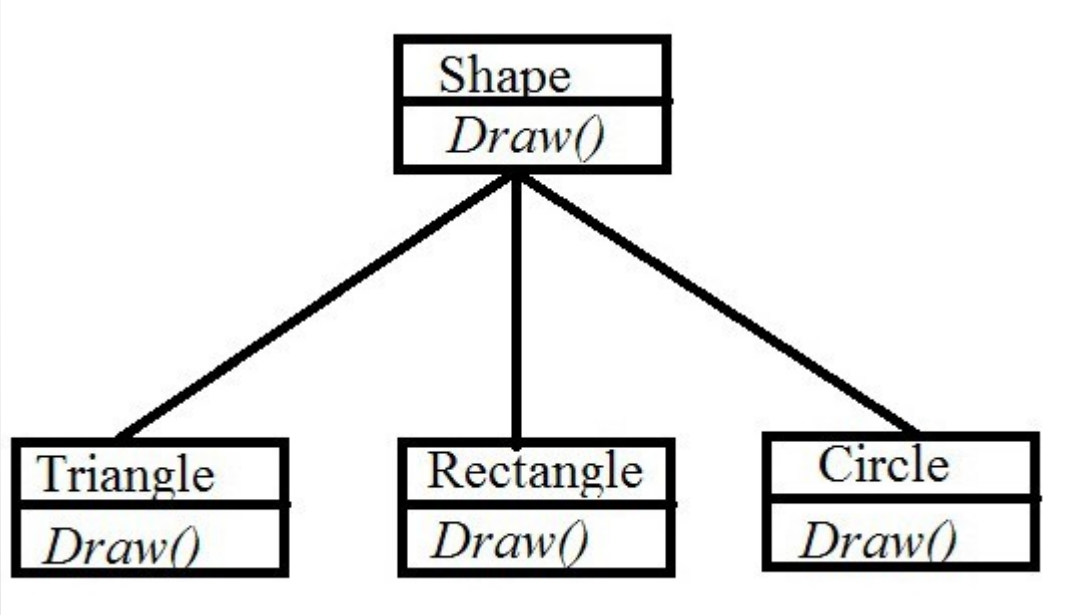
- **3)Data abstraction:** It refers to showing only the essential features of the application and hiding the details from outside world.
- **4)Data encapsulation:** It binds the data and functions together and keeps them safe to avoid outside interference and misuse.
- Encapsulation are implemented through the declaration of a class.
- A class contain **private,protected** and **public** members.
- By default all items defined in a class are **private**(Here members declared in this section are not visible outside the class). The members declared as protected are visible to its derived class but not outside the class.

- **5)Modularity:**
- Modularity is the process by which a larger program is sub-divided into smaller programs called modules.
- These modules are later linked together to build the complete software. These modules communicate with each other by passing messages.

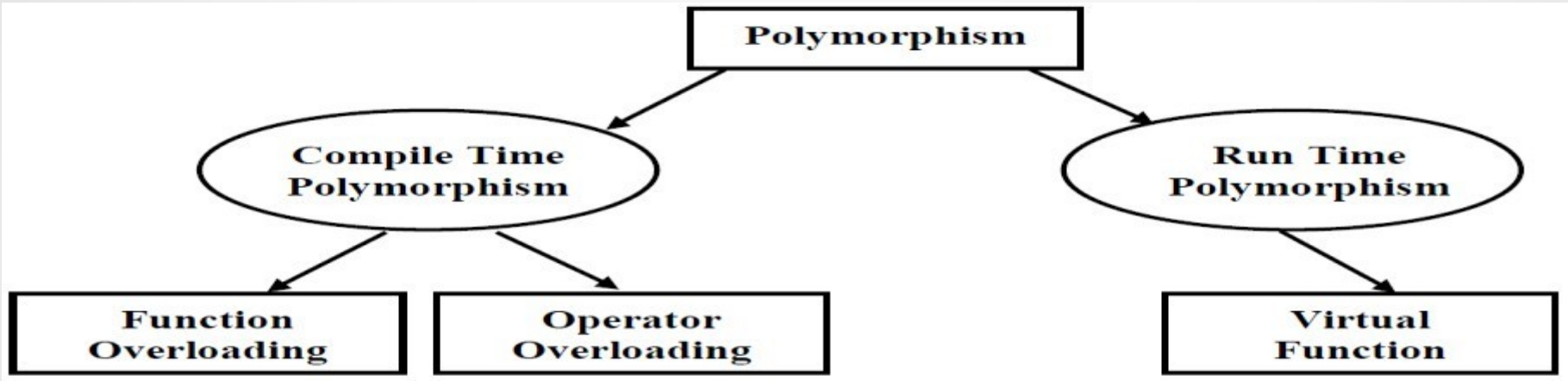




- **6)Polymorphism:** The ability to process objects differently depending on their data type or class.
- ‘ Poly’ means many,’ Morph’ means shape.
- Polymorphism is the ability of an objector function to take multiple forms.



Types of Polymorphism



(1) Compile time (Static) polymorphism (or early binding): Polymorphism during and compilation.

- Function overloading and operator overloading are examples.
- (2) Run time (Dynamic) polymorphism (or late binding): Polymorphism during run-time. It uses pointers.
- Virtual functions are examples.

- Function Overloading: Functions with the same name, but different signatures can act differently.
- Eg: The function prototypes int area (int, int); and int area(int); show that area() is an overloaded function.
- Operator overloading: It is the process of giving new meaning to an existing C++ operator.

Example -Operator overloading

```
#include <iostream>
using namespace std;
class TestClass
{
	private:
		int count;
	public:
		TestClass() : count(5) {}
		void operator --()//Overload the meaning of the -- operator.
		 {
			count = count – 3;//Decrement value by 1
		}
		void Display()
		{
				cout << "Count: " << count;
		}
};
```

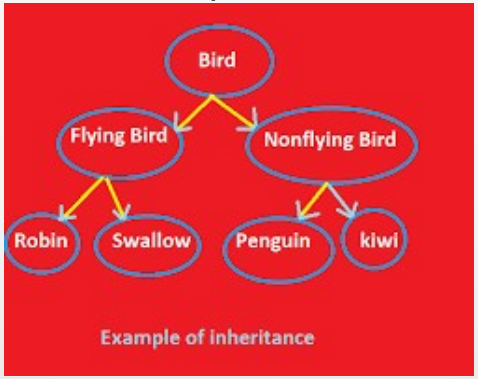
```
int main()
{
	TestClass tc;//Create an instance of the class TestClass and give it the name tc.
	--tc;//decrement operaotr
	tc.Display();
	return 0;//The function must return value upon successful completion.
}
```

Output

Advantages of Object Oriented Programming

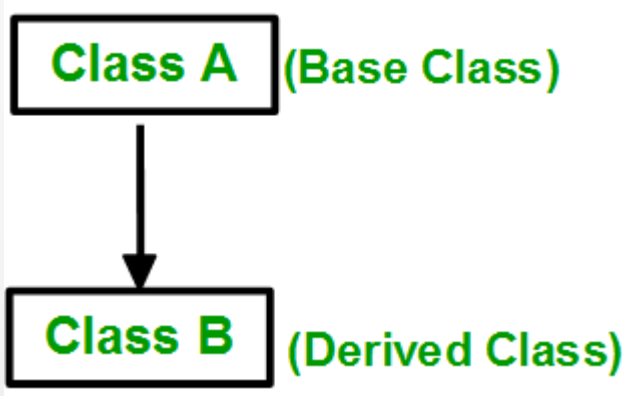
- Easy to maintain and modify code.
- Provides modular structure for programs.
- Code sharing.
- Information hiding.
- It is good for defining abstract data type.
- It implements real life scenario.
- It can define new data types as well as new operations for operators.

- **7)Inheritance:** It is the process by which objects of one class acquire the properties and behaviour of another class.
- The concept of inheritance provides re usability
- The existing class is called base class and the new class
- derived class (child) - the class that inherits from another class
- base class (parent) - the class being inherited from is called derived class.
- The different forms of inheritance are Single inheritance,Multiple inheritance,Multilevel inheritance,Hierarchical inheritance and Hybrid inheritance



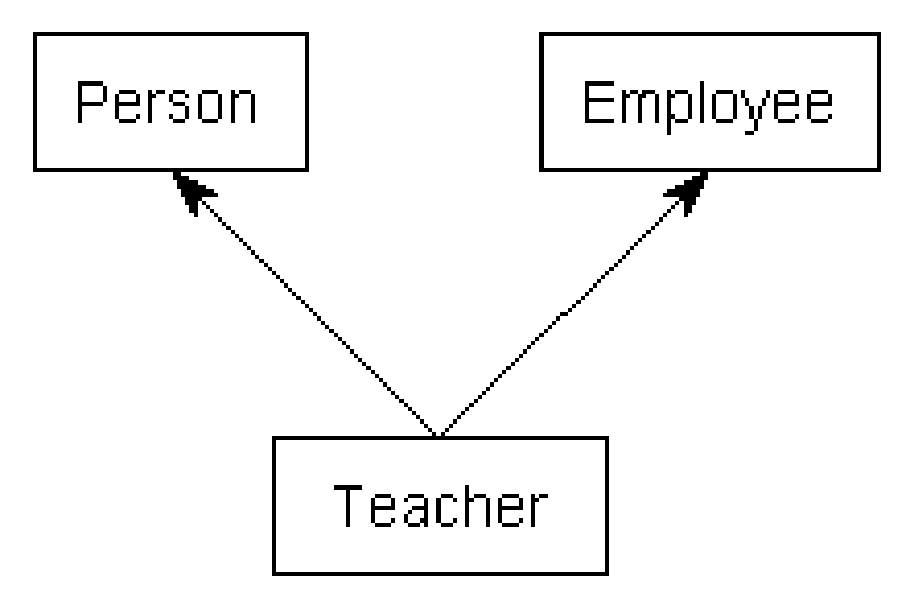
Single Inheritance

- Single inheritance is one type of inheritance in which the derived class inherits only one base class.
- It provides re usability by allowing the derived class to inherit the features of the base class using objects. A class whose properties are inherited for re usability is called parent class or superclass or base class.



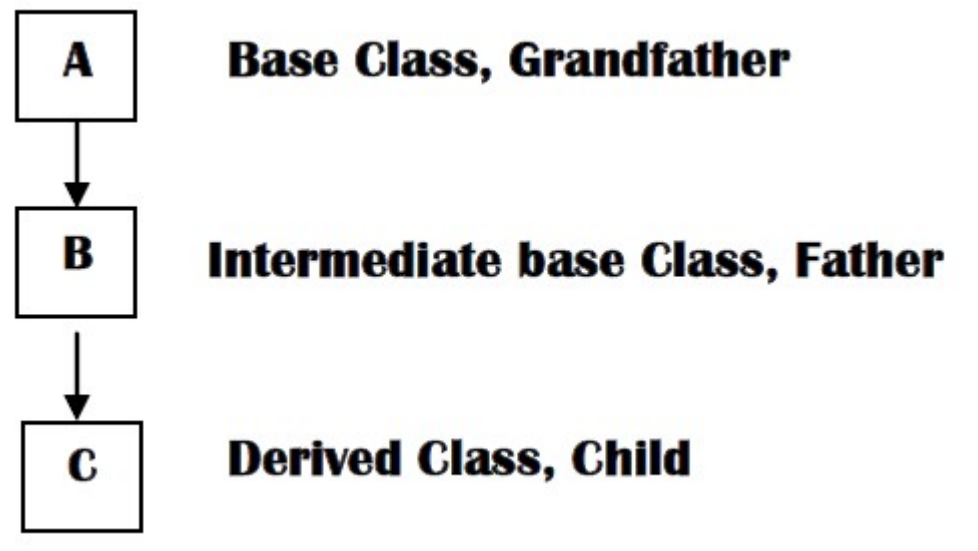
Multiple inheritance

- When a derived class inherits properties and behaviours of more than one base class, it is called multiple inheritance.
- In following figure, Teacher is a derived class from two base classes: Person and Employee.



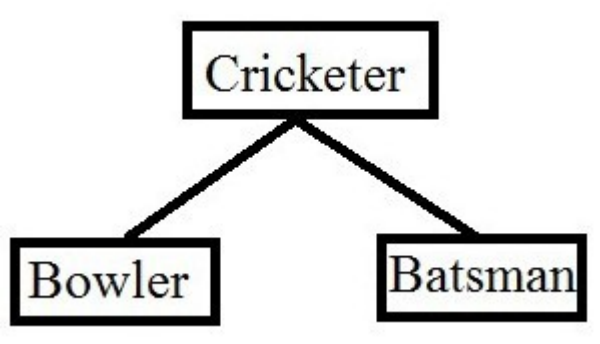
Multilevel Inheritance

- When properties and methods of a derived class are inherited by another class, it is called multilevel inheritance



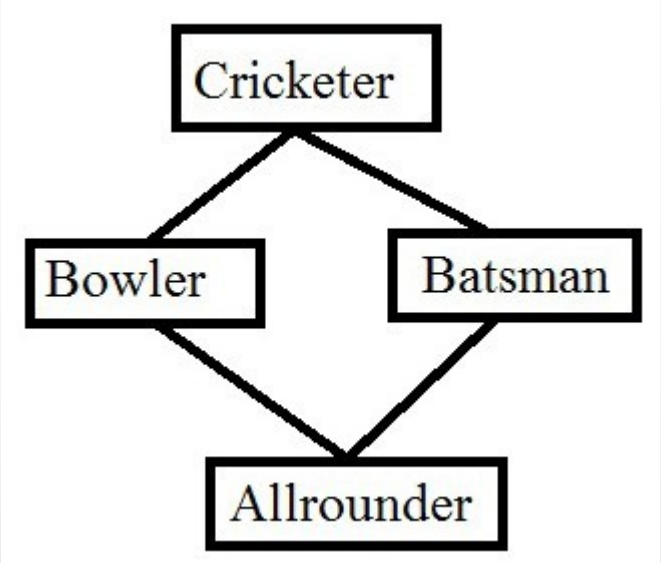
Hierarchical Inheritance

- When properties and behaviours of one base class are inherited by more than one derived class, it is called hierarchical inheritance.
- In following figure, Bowler and Batsman are two derived classes from same base class Cricketer.



Hybrid Inheritance

- Hybrid inheritance is a combination of multiple inheritance and multilevel inheritance.
- A class is derived from two classes as in multiple inheritance. However, one of the parent classes is not a base class. It is a derived class.



Procedural paradigm V/s OOP

Procedural paradigm	Object Oriented Paradigm
Data is undervalued.	Data is given importance.
Procedure is given importance.	Procedure is driven by data.
Creating new data types is difficult.	New data types and associated operations can easily be defined
Poor real world modelling.	Easy to define real world scenarios.
Employs top-down approach.	Employs bottom-up approach
Programs are decomposed into functions	Programs are decomposed into objects.

Thank You

Prepared By
Siraj.F.M
HSST Computer Science
AKGS GHSS Peralassery