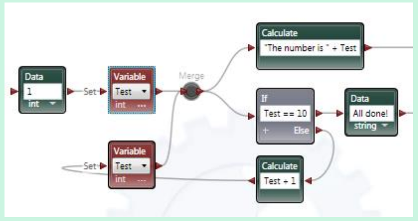
Introduction to Object Oriented Programming

Dept. of Computer Science & Engineering

Computer Programming

- A **computer program** is a step-by-step set of instructions for a computer
- **Programming Language** allow programmers to write a program that can perform a specific task
- With time and progress in technology, writing a program became much easier
- A low level language are often described as machine oriented languages that closer to the hardware
- High-level languages represent a giant leap towards easier programming
- There are five generations of programming language

Generation of Programming Language

Generation	Explanation	Example of Code
1st generation (Machine Language)	Uses a series of binary digits (1s and 0s)	10001110
2nd generation (Assembly Language)	Programmer writes instructions using symbolic instruction codes that are meaningful abbreviations	ADD 1001, 1100
3rd generation (Procedural Language)	Make complex programming simpler and easier to read, write and maintain	int x=10+5;
4th generation (Non-procedural Language)	Enables users to access data in a database	SELECT id FROM student
5th generation (Visual or Natural Language)	Provides a visual or graphical interface, called a visual programming environment, for creating source codes	 Calculate "The number is " + Test Data 1 int Set Variable Test int Merge If Test == 10 Else Data All done! string Set Variable Test int Calculate Test + 1

1st and 2nd generation are considered as low-level language
and the rest are considered as high-level language

Programming Approaches

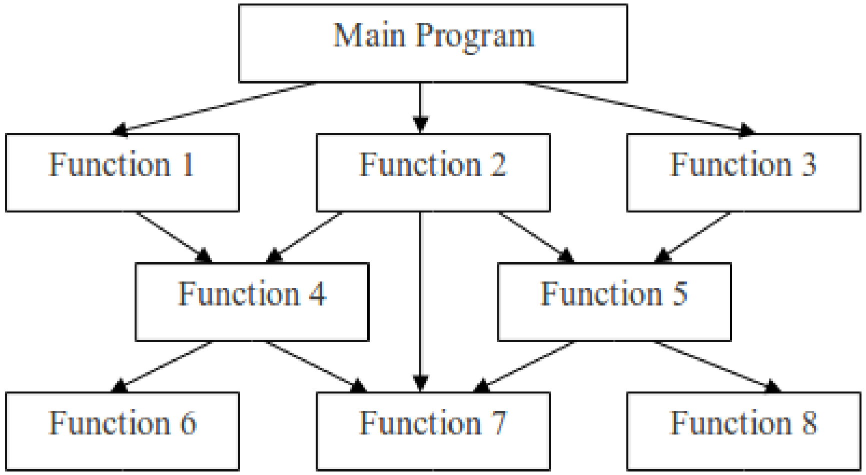
- Since the invention of the computer, many programming approache s have been tried.
  - Structured Programming
  - Modular Programming
  - Object Oriented Programming
  - Top-down & Bottom-up Programming

- The primary motivation in each case has been the concern to handle the increasing complexity of programs

Structured Programming

- Consists of writing a list of instructions for the computer to follow and organize these instructions into groups known as functions
- It also supports modular programming and follows top-down approach
- Usually functions share global data
- Global data are more vulnerable to change by a function
- In a large program it becomes difficult to identify which data is used by which function
- Can not model real-world problems very easily

Structured Programming (Contd.)



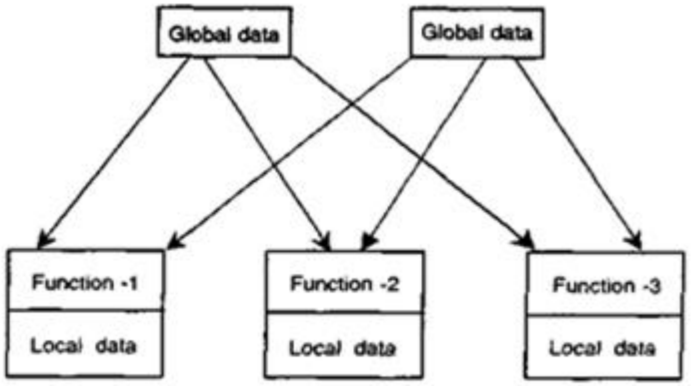


Fig. Organizations of data and functions in structured programming

Object Oriented Programming

- Object oriented programming is a programming approach that relies on the concept of classes and objects
- It removes some of the flaws encountered in the procedural approach of programming
- OOP treats data as a critical element and does not allow data to flow freely around the system
- Programs are divided into objects where the data and functions are built around these objects
- Objects may communicate with each other via functions
- New data and functions can be added whenever necessary

Object Oriented Programming (Contd.)

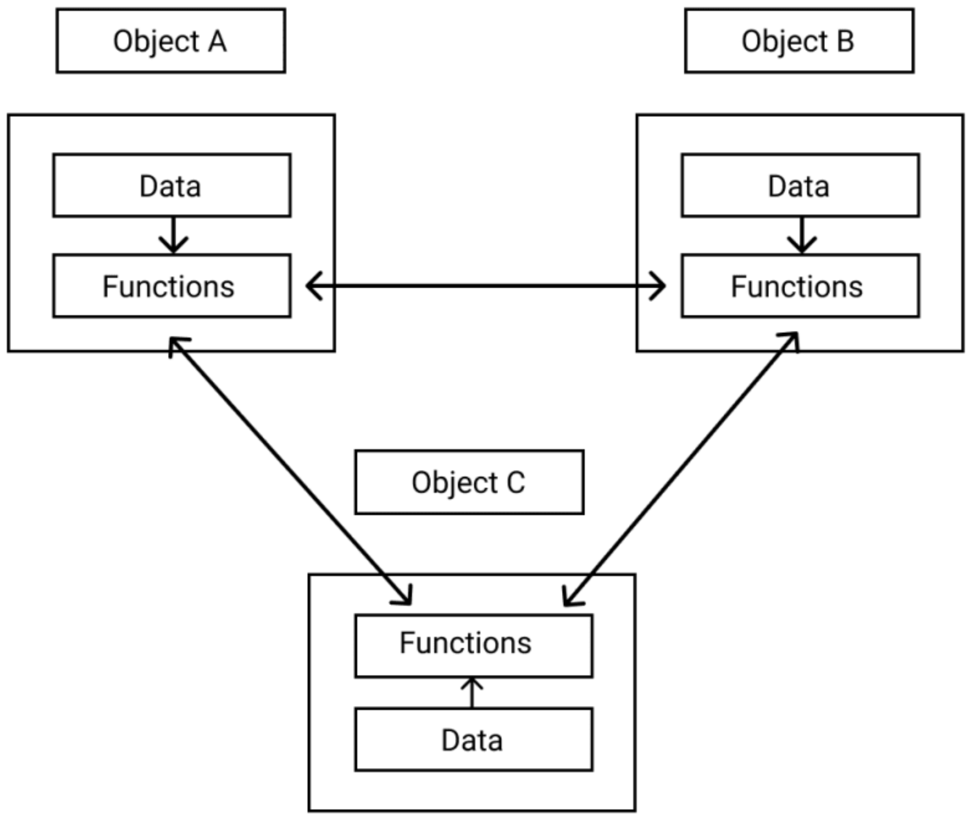


Fig. Organizations of data and functions in OOP

Structured vs Objected-oriented Programming

Structured Programming	Objected Oriented Programming
i) Program is divided into a number of sub-modules or functions	i) Program is organized by having a number of classes and objects
ii) It follows top-down approach	ii) It follows bottom-up approach
iii) No encapsulation. Data and functions are separate	iii) Allow encapsulation. Data and functions are put together
iv) Software reuse is not possible	iv) Software reuse is possible
v) Example: ALGOL, PASCAL, C	v) Example: Java, Python, C++

OOP Basic Concepts

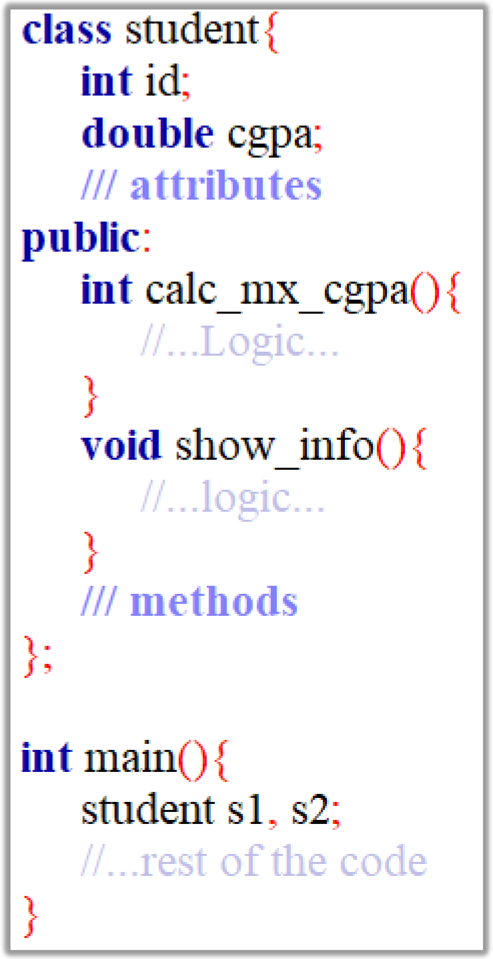
- **<u>Object</u>:** Represents entity in an object-oriented system. It can be person, place, bank account or a table of data etc.
- Object is anything to which you can apply a concept
- Object contains data and methods

Student Car Bank account

Fig: Example of objects

OOP Basic Concepts (Contd.)

- **Class:** A category of similar objects
- Class act as a blueprint for creating object
- Class consists of three things: class name, attributes and methods
- Attributes are properties of an object
- Methods represents actions performed by the object



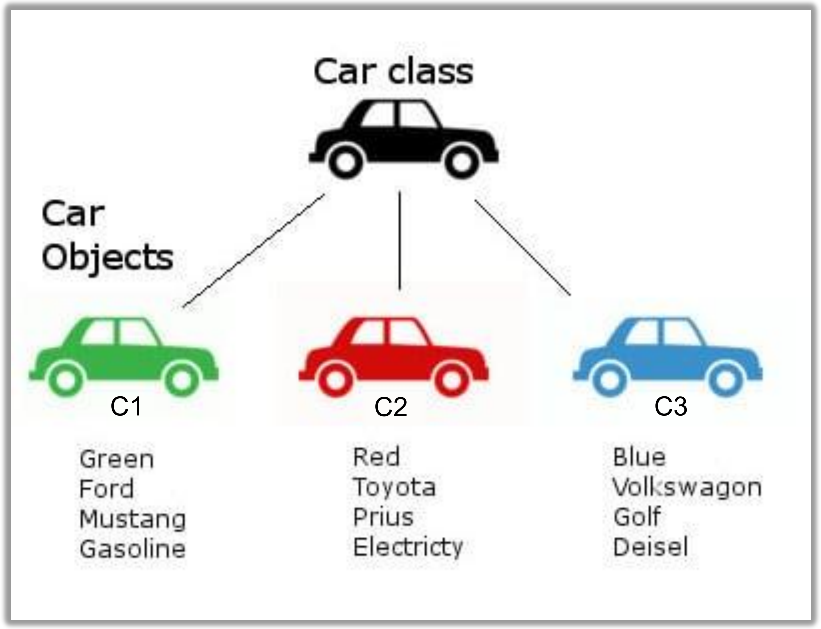
```
class student{
      int id;
      double cgpa;
      /// attributes
public:
      int calc_mx_cgpa(){
            //...Logic...
      }
      void show_info(){
            //...logic...
      }
      /// methods
};

int main(){
      student s1, s2;
      //...rest of the code
}
```

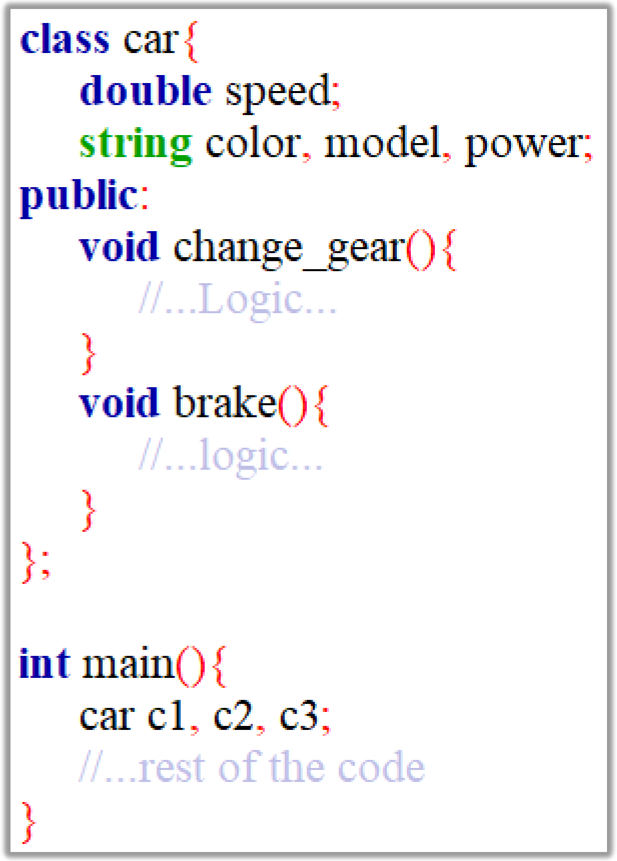
In the code, student is the class name

In the main function,
two objects s1 and s2 have been created

OOP Basic Concepts (Contd.)

- **Another example of class:** A vehicle class containing some attributes and methods





```
class car{
    double speed;
    string color, model, power;
public:
    void change_gear(){
        //...Logic...
    }
    void brake(){
        //...logic...
    }
};

int main(){
    car c1, c2, c3;
    //...rest of the code
}
```

Each copy of an object from a particular class is called an instance of the class.

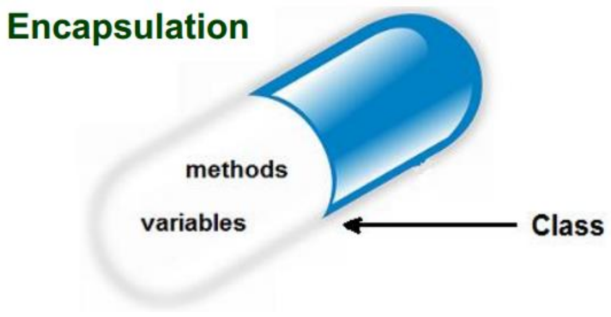
Class vs Object

Class	Object
i) Class is a data type	i) Object is an instance of Class
ii) It generates OBJECTS	ii) It gives life to CLASS
iii) Does not occupy memory location	iii) It occupies memory location
iv) It cannot be manipulated because it is not available in memory	iv) It can be manipulated during runtime

Basic Features of a Class

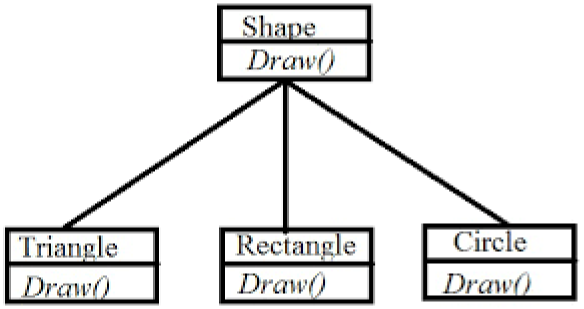
- All OOP languages, including C++, share three common features
  1) Encapsulation
  2) Polymorphism
  3) Inheritance

- **Encapsulation**: It is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse
- It wraps data and functions into a single unit



Basic Features of a Class (Contd.)

- **Polymorphism**: Polymorphism is the quality that allows one name to be used for two or more related but technically different purposes
- Polymorphism means "many forms"
- In C, to find the absolute value of integer, long and float data types we need to use abs(), labs() and fabs() function name
- But in C++, we can do these for all three data types using abs() function name

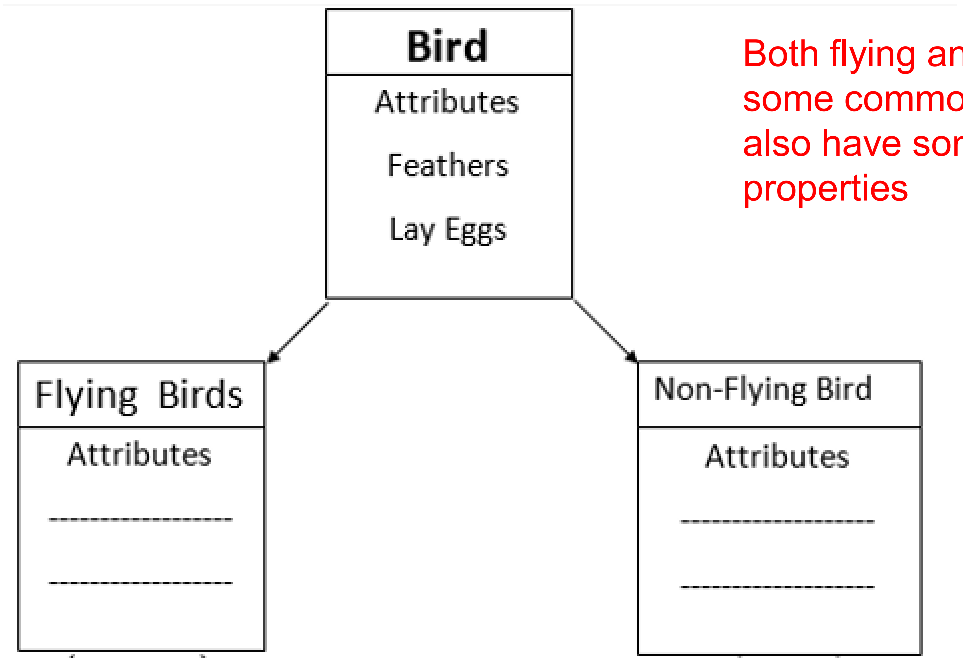


Here, depending on the parameter it will draw the shape

Fig: Drawing a shape using same function name

Basic Features of a Class (Contd.)

- **Inheritance**: Inheritance is the process by which one object can a cquire the properties of another
- An object can inherit a general set of properties from a class and can also add extra features that are specific only to itself



Both flying and non-flying birds have some common properties, but they also have some distinguishing properties

Benefits of Using OOP

- OOP allows multiple instances of an object to co-exist without any interference
- Encapsulation ensures that the data is protected from outside function blocks
- Through inheritance, we can eliminate redundant code
- OOP systems can be easily upgraded from small to large systems
- Software complexity can be easily managed

Main advantage is that, you can create an object which represents some real one, put the logic inside it and hide all the implementation details behind some interface

Thank You