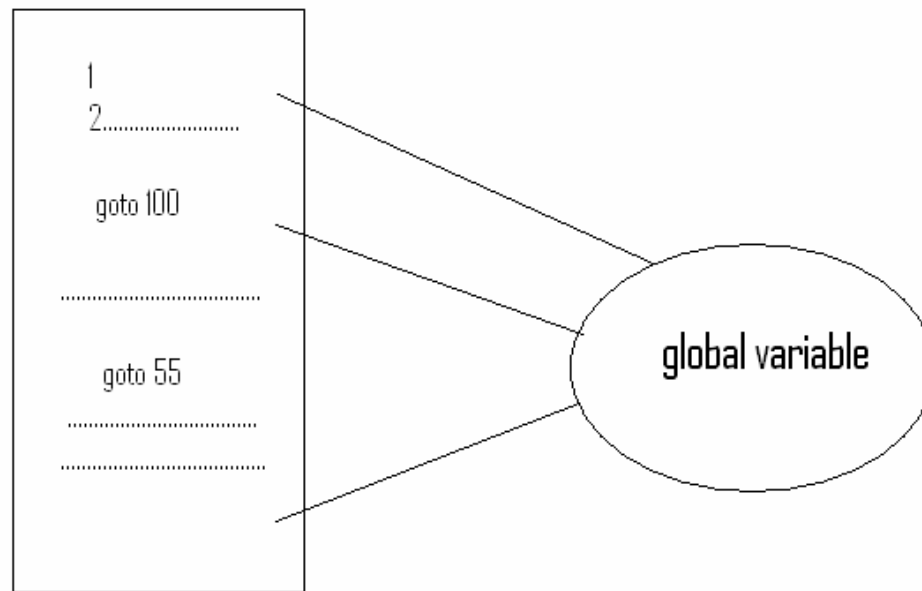


Object Oriented Programming

Evolution Of Programming Paradigms

- Monolithic programming
- Procedural programming
- Structural programming
- Object oriented programming

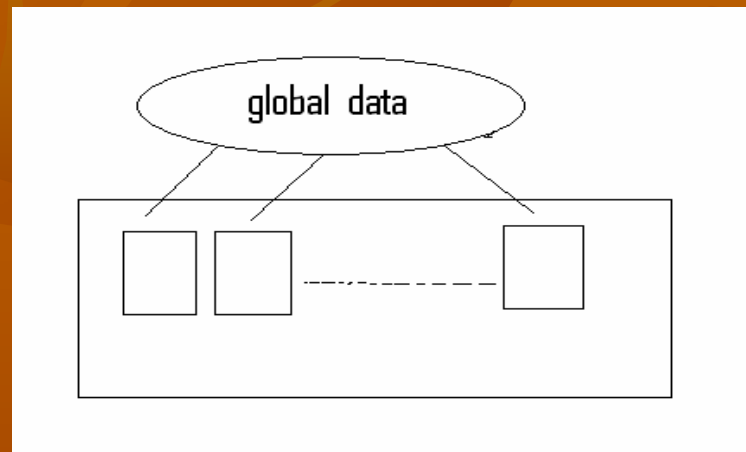
Monolithic programming



Difficulty:

Debugging

Procedural programming

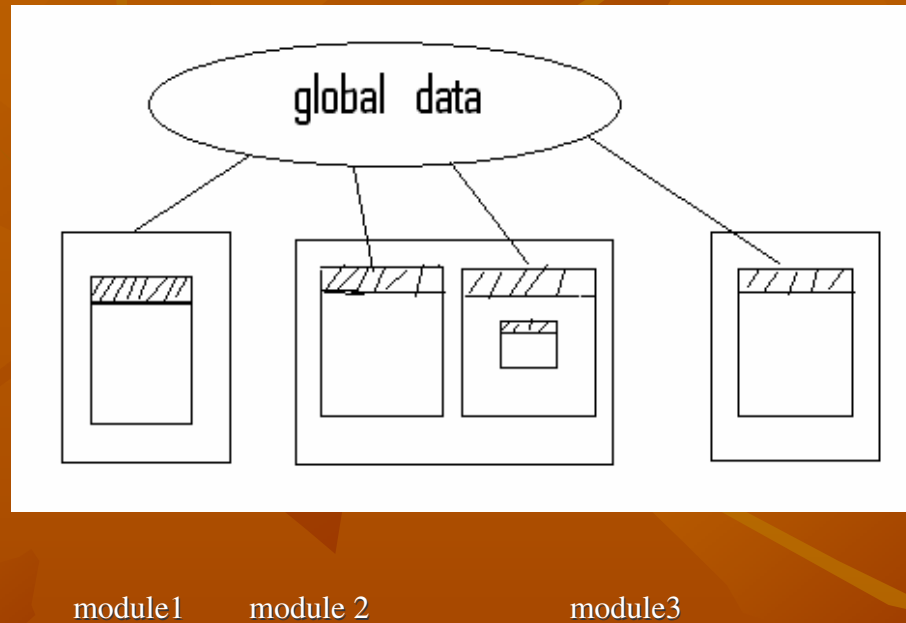


1. Programs are organized in the form of subroutines and all data items are global
2. Programs control is through jumps (goto's) and call to subroutines.

Ex : FORTRAN & COBOL

Difficulty: In debugging

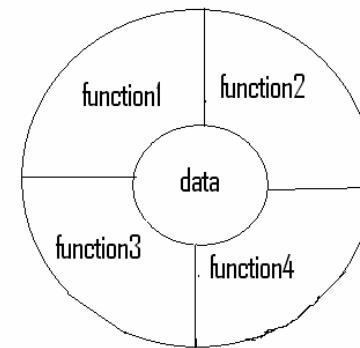
STRUCTURAL PROGRAMMING



- Emphasis on algorithm rather than data
- Procedures have their own local data
- Controlled scope of data
- A rich set of control structures to avoid goto's
ex: PASCAL & C

OBJECT ORIENTED PROGRAMMING

Objects = data + functions



Classes: objects with the same behaviour (operations) are grouped into a class

Objects: Instances of a class

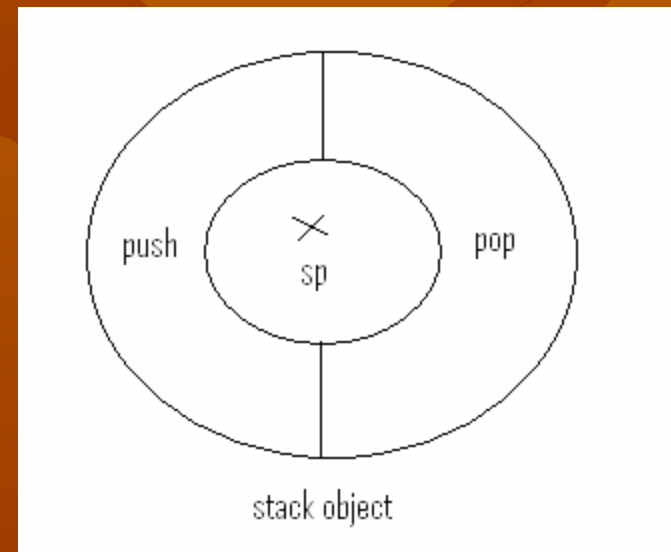
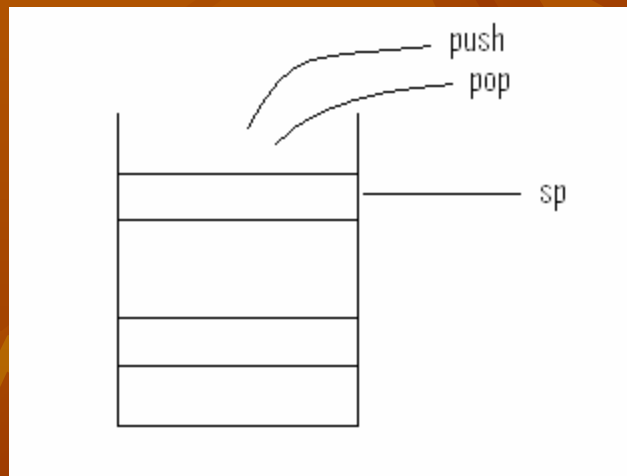
ex: fruit: apple, orange,..... are objects

Objects are specialization.

Classes are generalization.

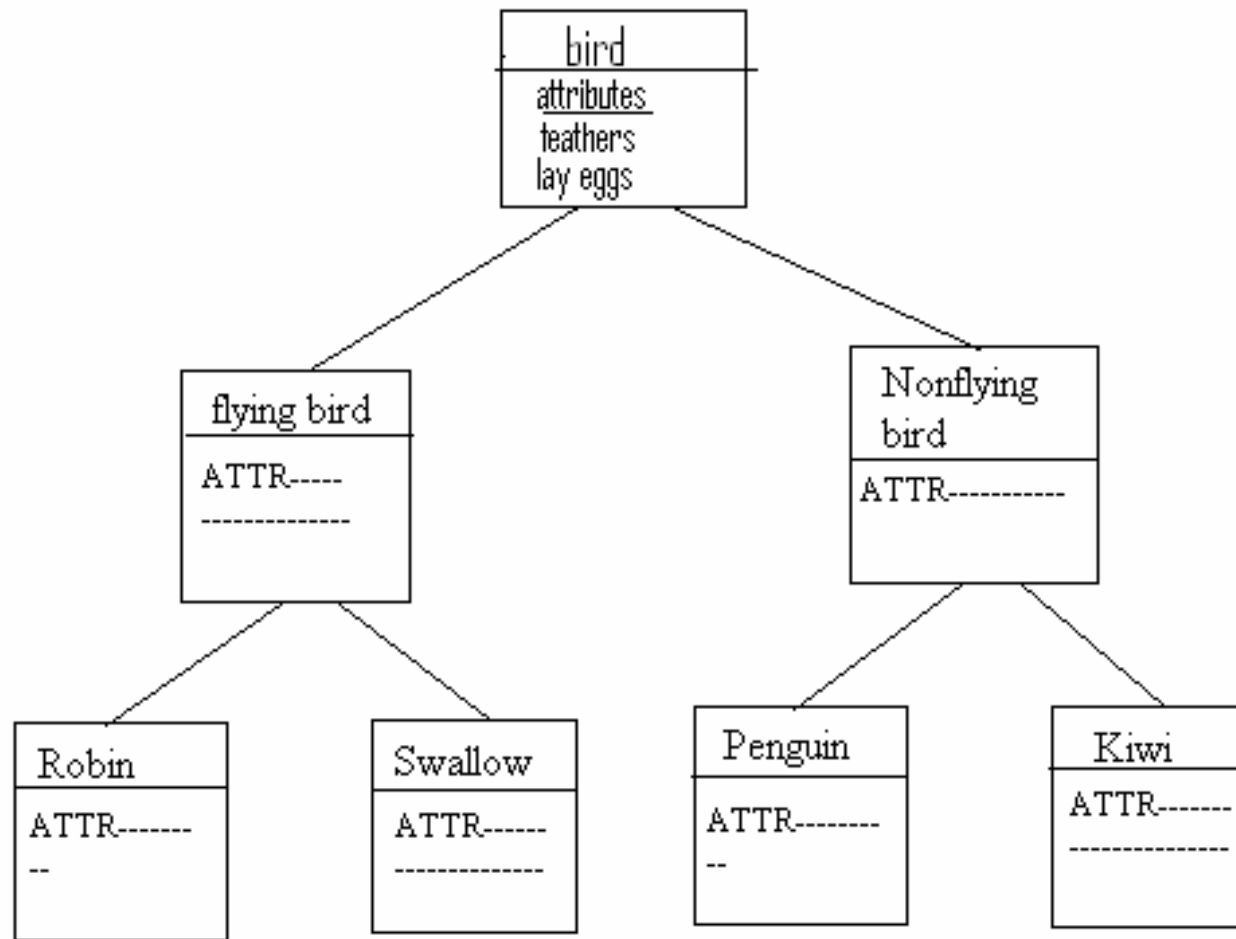
■ Important features:

1. Data hiding



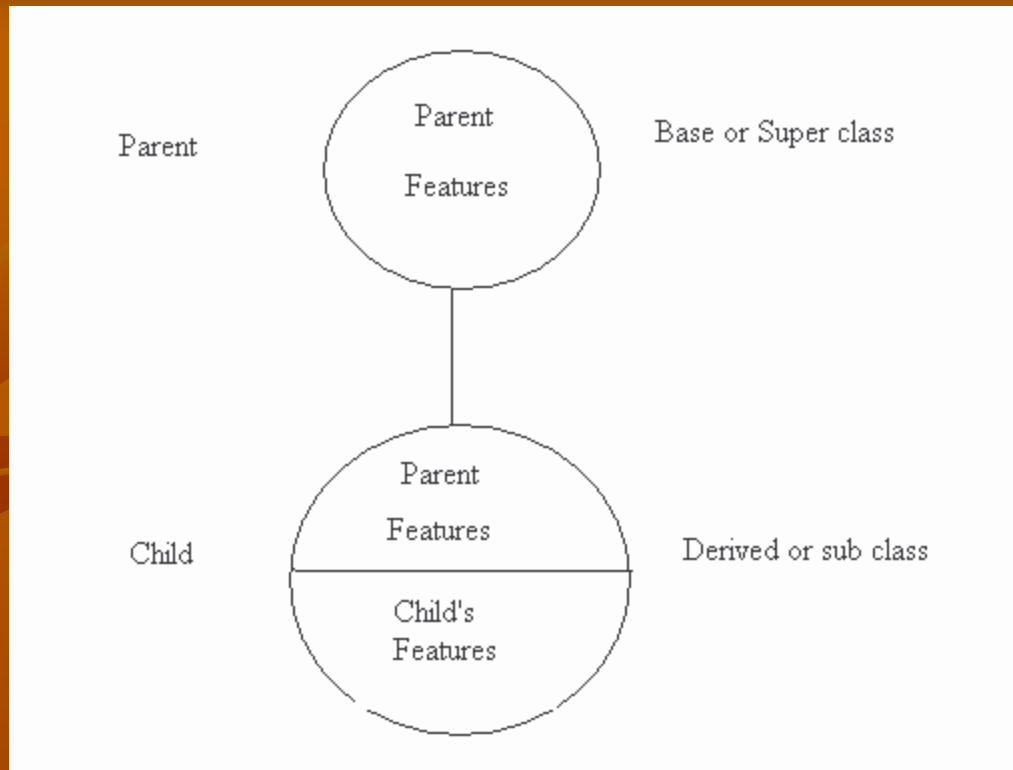
X- Can be accessed only through push & pop functions.

■ Inheritance:



Specializing



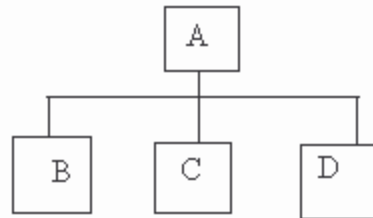


Inheritance relation is often called *is-a* relation.

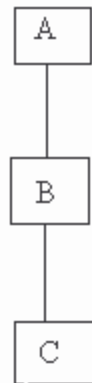
Types of inheritance:



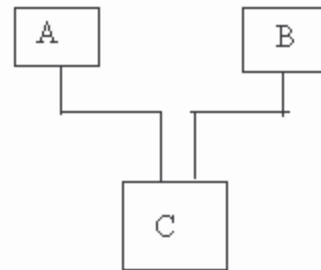
Single Inheritance



Hierarchical Inheritance



Multilevel Inheritance



Multiple Inheritance

Polymorphism

- Meaning: Ability to take more than one form
- Operator: ‘+’

2+3=5 (scalar addition)

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}$$

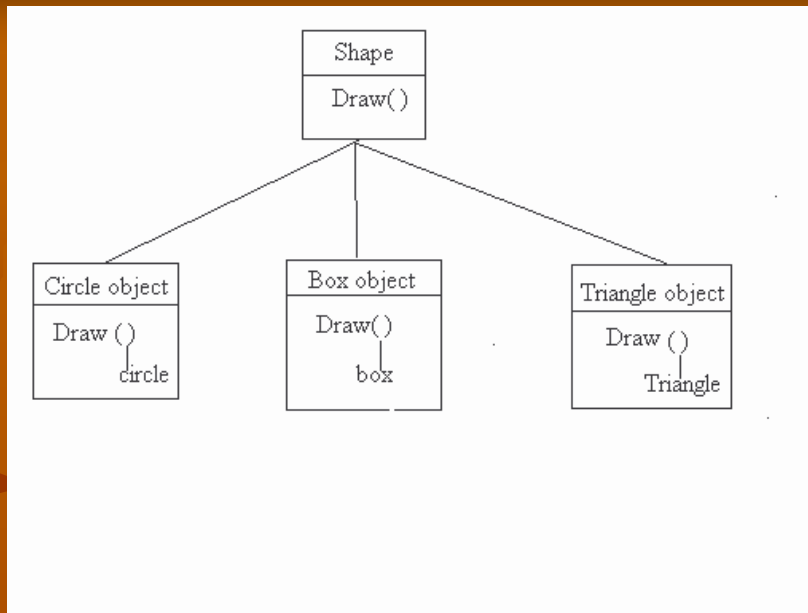
(vector addition)

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + \begin{pmatrix} 3 & 2 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 4 & 4 \\ 4 & 8 \end{pmatrix}$$

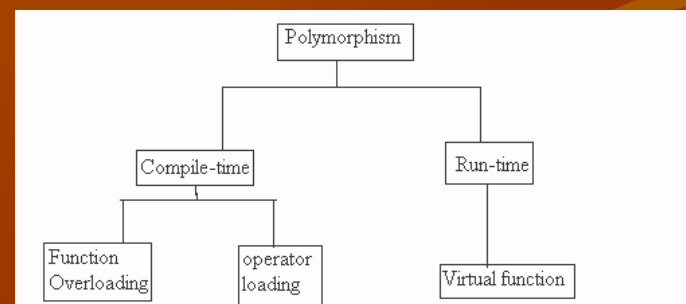
(Matrix addition)

“pot” + “pole” = pot hole (string concatenation)

operator is fixed. But operands are changing . The process of making an operator to exhibit different behavior in different instances is known as “ operator – overloading”.



The same function draw() can be used to handle different types of arguments: This is known as function overloading.



Dynamic Binding

- This refers to the linking of a procedure call to be executed in response to the call. when the code associated with the given procedure call is not known until the time of the call at run time we have to opt for dynamic binding

Message Passing

- In OOP we follow the following basic steps:
 1. Creating classes that define objects and their behavior
 2. Creating objects from class definition
 3. Establishing communication among objects

Message passing involves specifying the name of the object, the name of the function (message) and the information to be sent.