



AISSMS

INSTITUTE OF INFORMATION TECHNOLOGY

ADDING VALUE TO ENGINEERING



An Autonomous Institute Affiliated to Savitribai Phule Pune University
Approved by AICTE, New Delhi and Recognised by Govt. of Maharashtra
Accredited by NAAC with "A+" Grade | NBA - 5UG Programmes

Unit 1-Concepts of Object Oriented Programming

Subject: Object Oriented Programming

Dr. Meenakshi A. Thalor

Ph.D.(Computer Engineering),M.E.(I.T.),B-Tech(I.T.)

Head of Department

Information Technology ,AISSMS IOIT, Pune

Teaching Scheme

Second Year Information Technology (2022 Course) Object Oriented Programming				
Course Code:	ITPCC403	Credit:	3	
Contact Hrs.:	3 Hrs/week (L)	Type of Course:	Lecture	
Examination Scheme	In-sem. Evaluation 40 Marks	Practical/Activity based Evaluation 60 Marks		

Second Year Information Technology (2022 Course) Object Oriented Programming Lab				
Course Code:	ITPCC407	Credit:	2	
Contact Hrs.:	4 Hrs/week (P)	Type of Course:	Practical	
Examination Scheme	Term-work 25 Marks	Oral 25 Marks		

Course Objective and Outcome

Course Objectives

- | | |
|---|---|
| 1 | To explain object-oriented programming concepts |
| 2 | To create classes and objects for real world problems |
| 3 | To use inheritance and polymorphism |
| 4 | To create abstract class and Interface |
| 5 | To handle exceptions using exception handling |
| 6 | To make use of collections for implementing generics |

Course Outcomes: Students will be able to

- | | |
|-------|---|
| 403.1 | Describe the object-oriented programming paradigm |
| 403.2 | Define classes and objects for real world problems |
| 403.3 | Implement inheritance and polymorphism for given problems |
| 403.4 | Implement abstract class and interface |
| 403.5 | Handle exceptions using exception handling mechanism |
| 403.6 | Make use of collections for implementing generics programming |

UNIT-I

CONCEPTS OF OBJECT-ORIENTED PROGRAMMING

(6 Hrs.)

Introduction to Procedural-Oriented and Object-Oriented Paradigms, Limitations of Procedural Programming, Need of Object-Oriented Programming, Fundamentals of the Object-Oriented Programming (OOP): Objects, Classes, Data abstraction, Encapsulation, Inheritance, Polymorphism, Dynamic Binding and Message Passing, Data types, Control structures, Arrays and Strings, Case study

Case Study: Model a real world scenario (vehicle class, fruit class, student management in university etc.) using Object Oriented Paradigm

Basic Terminology

- **Instruction:**

An **instruction** is a single operation of a processor

- **Program:**

A **computer program** is a collection of **instructions** that can be executed by a **computer** to perform a specific task. A **computer program** is usually written by a **computer** programmer in a **programming** language.

- **Software:**

Software comprises the entire set of **programs**, procedures, and routines associated with the operation of a computer system.

Software Evolution

- Computers can understand only when the information is given in a language that the computer can understand. In order to make the man-machine communication easier the languages are developed.

Machine Code

```
01010101011
01010101010
11010101010
```

Assembly Language

```
SUB AX , BX
MOV CX , AX
MOV DX , 0
```

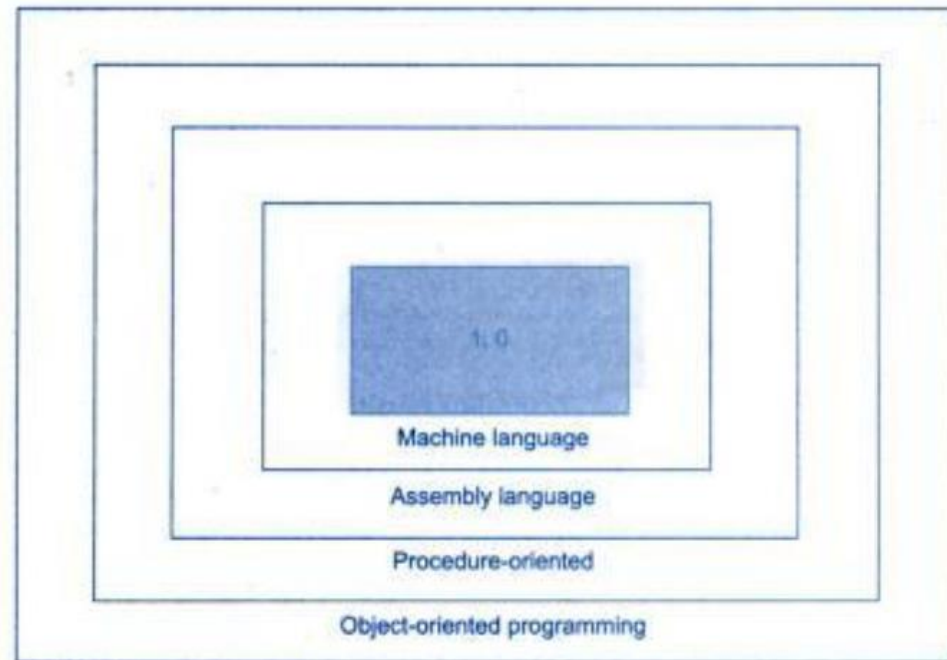
```
print("Program To Add Two Numbers : ")
numberone = int(input("Enter Number One : "))
numbertwo = int(input("Enter Number Two : "))

total = numberone + numbertwo
print("Addition of Two Number is : ",total)

input("Press Any Key To Quit : ")
```

Software Evolution

- Software evolution has different phases or layers of growth. These layers are designed one over the last



Software Evolution

- Using any language we can write software instructions. With the help of software user can write high-level program. A program is nothing but a sequence of instruction that will be executed one after other. Software can be classified into two types:
 1. System software
 2. Application software
- **System software:** Directly interacts with the computer system. Operating system, compiler, interpreter are examples for this.
- **Application software:** All the programs written by a user with the help of any software is called as application software. Eg. Balance sheet preparation for a company, monitoring rail way reservation process etc.,

Introduction to Programming Languages

- The shift in programming language is categorized as following:

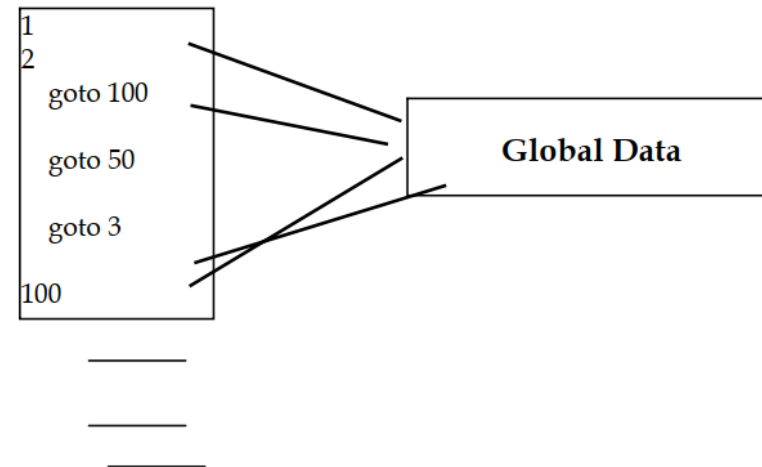
1. Monolithic Programming
2. Procedural Programming
3. Structural Programming
4. Object Oriented Programming

Monolithic Programming

- This programming consists only global data and sequential code. Program flow control is achieved through the use of jump and the program code is duplicated each time it is used.
- No subroutine concept is used. Since this programming style is not supporting the concept of data abstraction it is very difficult to maintain or enhance the program code

E.g.-

- **Assembly language**
- **BASIC**



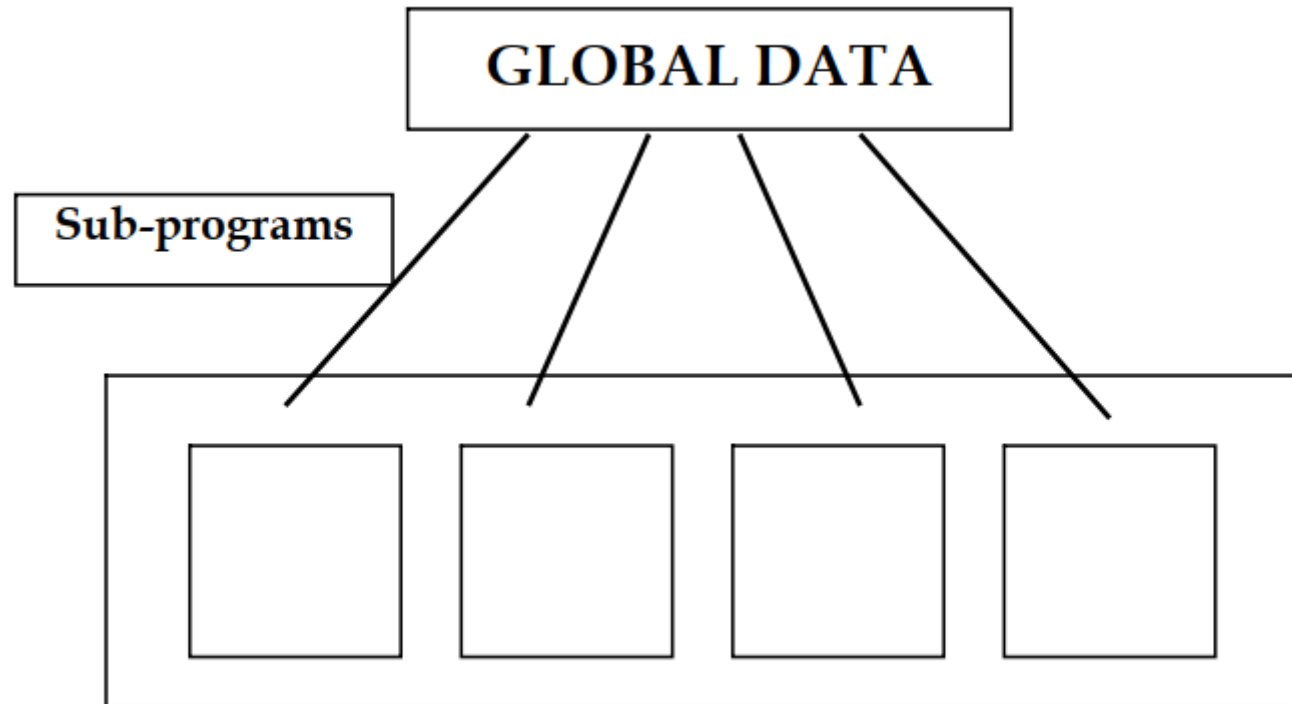
Limitation of Monolithic Programming

- The data is global and code operates on it
- As the size of code increases, maintenance is a problem
- Does not have independent data for processing
- The concept of local variables did not exist
- Reusability of code was not supposed

Procedural Programming

- The important features of Procedural Programming are
 1. Programs are organized in the form of subroutines and all data items are global
 2. Program controls are through jumps (goto's) and call subroutines
 3. Abstracted subroutines are used to avoid repetition
 4. Software application is minimized
 5. Difficult to maintain and enhance the program code
- E.g.-
 - Fortran
 - COBOL

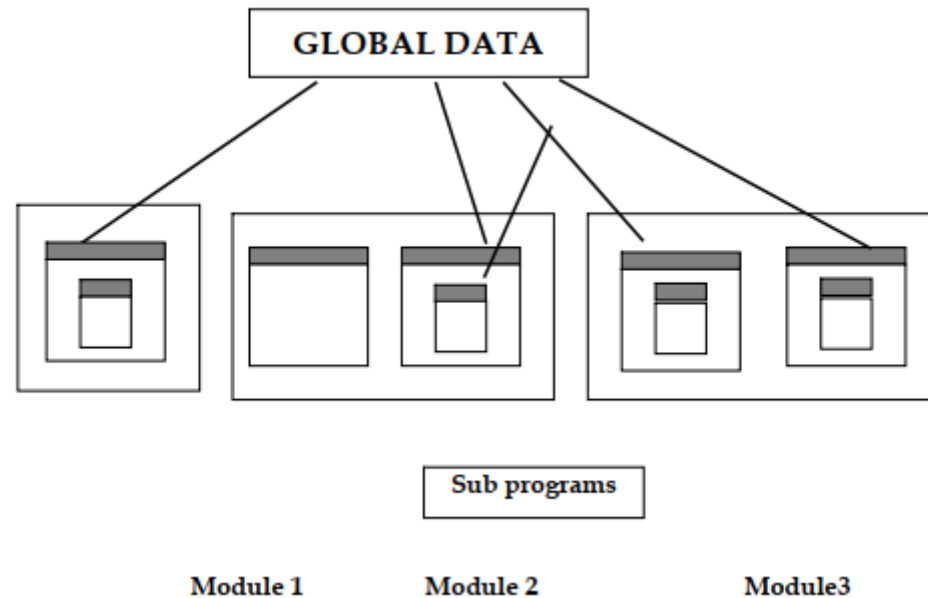
Procedural Programming



Structured programming

- Programs consist of multiple and in turn each module has a set of functions of related types.
 1. Structured programming is based upon the algorithm rather than data
 2. Programs are divided into individual modules that perform different task.
 3. Controls the scope of data
 4. Support modular programming
 5. Introduction of user defined data types
- E.g.-
 - Pascal
 - C

Structured programming



Object Oriented Programming

- The object oriented programming has taken the best ideas of structured programming and combined them with several powerful concepts that encourage us to approach the task of programming in a new way. An **object** is a combination or collection of data and code designed to emulate a physical or abstract entity. Each object has its own identity and is distinguishable from other objects.

- **E.g.-**

- C++
- Smalltalk
- Eiffel
- Java

Object Oriented Programming

- Depending on the object features supported, the languages are classified into two categories:
- Object-Based Programming Languages
- Object-Oriented Programming Languages
- Object-based programming languages support encapsulation, object identity without supporting the important features of inheritance, polymorphism and message communications. Example Ada language. Major features that are required for object based programming are:
 1. Data encapsulation
 2. Data hiding and access mechanisms
 3. Automatic initialization and clear-up of objects
 4. Operator overloading

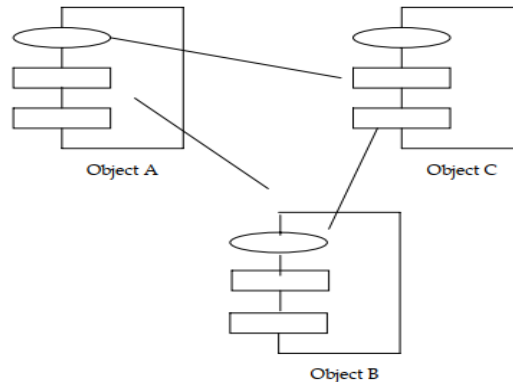
Object-based language = **Encapsulation** + **Object Identity**

Object Oriented Programming

- Object-Oriented Programming Language incorporate all the features of object-based programming languages along with inheritance and polymorphism.

Object-oriented programming language =

Object = based language + polymorphism + inheritance



Features of Object-Oriented Programming

1. Improvement over the structured programming languages.
2. Emphasis on data rather than algorithm
3. Data abstraction is introduced in addition to procedural abstraction
4. Data and associated operations are unified into a single unit, thus the objects are grouped with common attributes, operations and semantics.
5. Programs are designed around the data being operated, rather than operations themselves

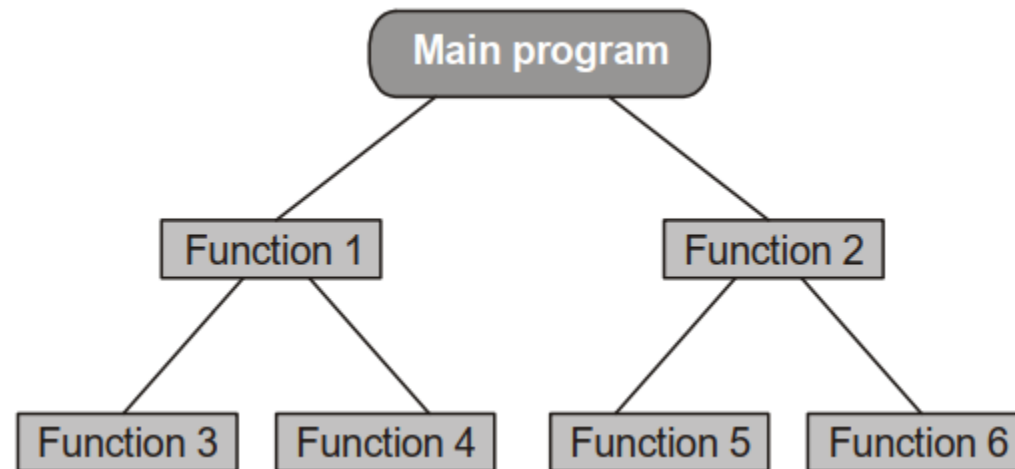
Generic Programming

- Generic programming refers to the programming model where algorithms and functions are written in terms of 'types' so that it works on all data types and not just for one.

```
#include <iostream>
using namespace std;
template <typename T>
void show(T data)
{
    cout<<data<<endl;
}
int main ()
{
    show(24);
    show(34.67);
    show("RAJ");
    return 0;
}
```

Overview of Procedure Oriented Programming

- Using the procedure oriented approach, the programmer view a problem as a sequence of things to do. The programmer organizes the related data items and write the necessary functions (procedures) to manipulate the data and the process, complete the sequence of tasks that solve the problem. The primary focus is on the functions.



Example

- For example: Consider a payroll system where employee pay-slip is to be generated. Payroll Program is the main program and Accept(), Calculate(), Display() are the functions.
 - Accept(): This function can be used to accept the employee details.
 - Calculate(): This function can be used for calculating the salary and the deductions.
 - Display(): This function can be used for displaying the employee pay-slip.

Features of Procedure-Oriented Programming

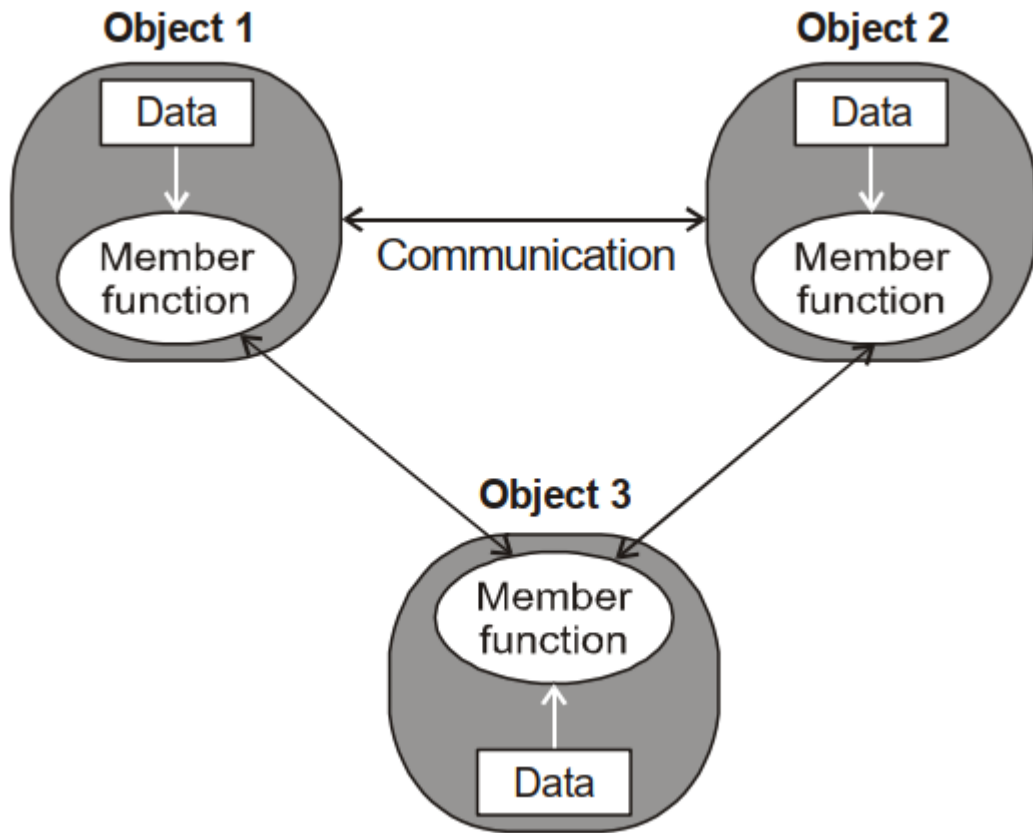
- Features of Procedure-Oriented Programming are as follows:

1. Focus is on the functions.
2. It follows Top-Down approach (while programming)
3. Program consists of different functions.
4. Most of the functions share global data.
5. Functions transform data from one form to another.
6. Data is not hidden and can be easily shared.

Object-Oriented Programming

- Object-Oriented Programming (**OOP**) contains the concepts of Procedure-Oriented Programming and also added some additional concepts. In OOP the main focus is on the data that is to be used rather than the function.
- Once the data, which is to be used, is decided then the different functions that will operate on this data are defined. Thus it follows Bottom-Up approach. OOP allows **decomposition** of a problem into a number of entities called as objects and then builds data and functions around these objects.

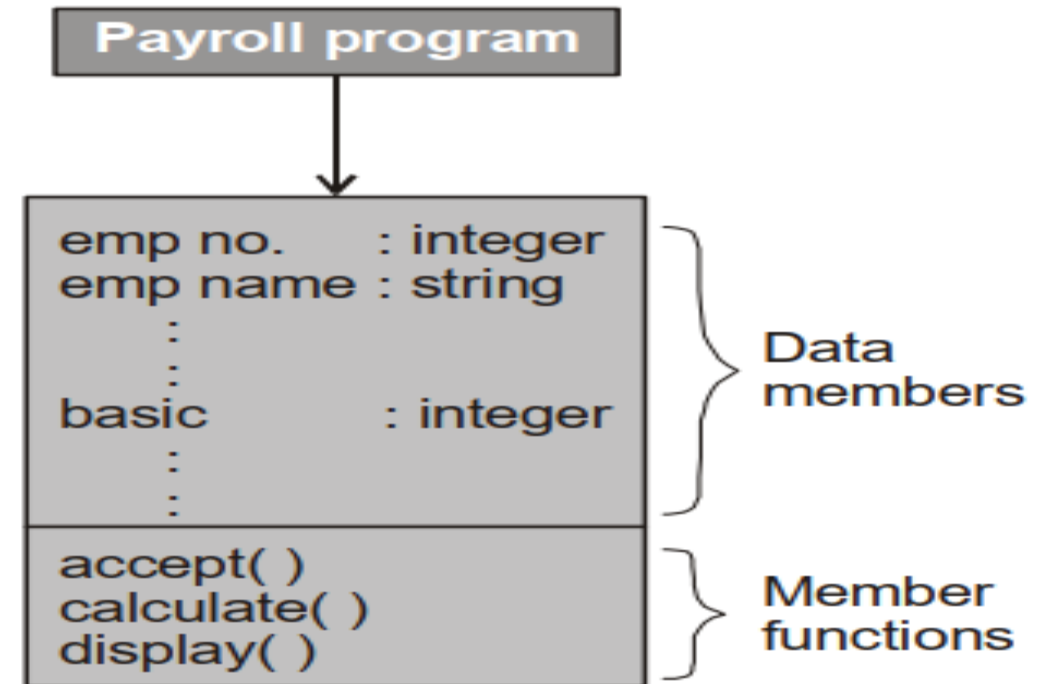
Object-Oriented Programming



1. Since the data is hidden it cannot be accessed directly by any outside function.
2. If one wants to access or modify the data of an object then he should know the different functions, which are associated with those objects.
3. Objects can pass messages to each other through functions.

Example

For example: Consider a payroll system where employee pay-slip is to be generated.



Characteristics of Object-Oriented Programming

1. Focus is on data rather than procedures or functions.
2. Program consists of different objects.
3. Data Structures are designed to characterize the objects.
4. In Data Structure functions that operate on the data of an object are tied together.
5. Data is hidden and can only be accessed through the object's member functions.
6. Objects can pass messages to each other through functions.
7. New data and Functions can be easily added whenever necessary.
8. It follows bottom-up approach.

POP Language Vs. OOP Language

	Procedure Oriented Programming	Object Oriented Programming
i.	Emphasis is on doing things (algorithm).	Emphasis is on data rather than procedure.
ii.	Large programs are divided into smaller program known as function.	Programs are divided into objects.
iii.	Most of functions share global data.	Functions that operates on the data of an object are tied together in the data structure.
iv.	Data move openly around the system from function to function.	Data is hidden and cannot be access by extend function.
v.	Function transform data from one form to another.	Objects may communicate with each other through function and new data and function can be easily added.
vi.	Employs top-down approach in program design.	Follows bottom-up approach in program design.
vii.	The unit in POP is function	Unit of OOP is class.
viii.	It concentrate on creating function.	It starts from isolating the classes and then look for methods inside them.

Concepts of OOP

- Object
- Classes
- Data abstraction
- Data encapsulation
- Polymorphism
- Inheritance
- Message passing

Objects

- An entity that has state and behavior is known as an object e.g. chair, bike, pen .It can be physical or logical. The example of logical entity is banking system.
- An object has three characteristics
 1. state- represent data of an object
 2. Behavior-Represent functionality of an object such as deposit, withdraw
 3. Identity- Identity implemented by unique id. The value of id is not visible to external user. But it is used by JVM to identify each object uniquely.

Example

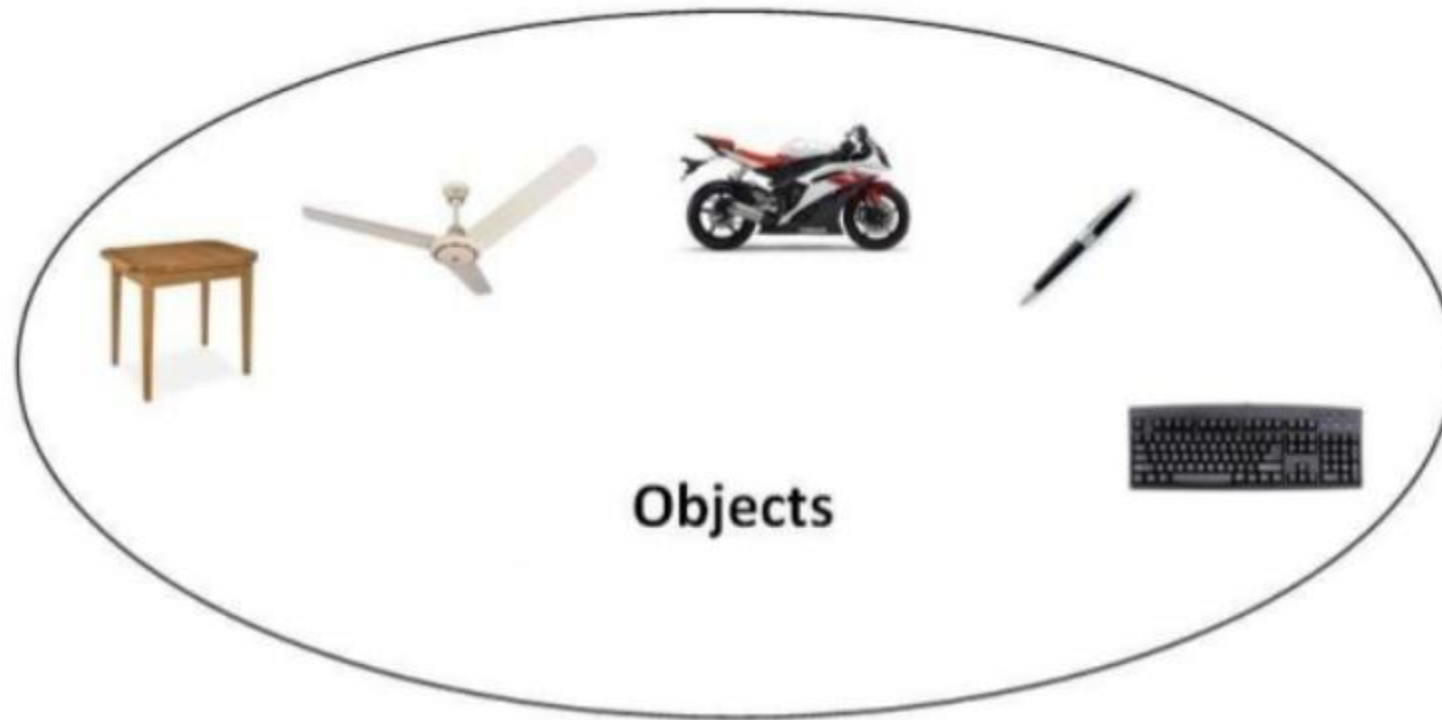
Example1:

- Object: house:
- State: Location,color,area
- Behaviour: close/open main door

Example2:

- Object:Car
- State: color,make
- Behaviour: Accelerate ,slowdown

Identify state and behavior of each?



Class

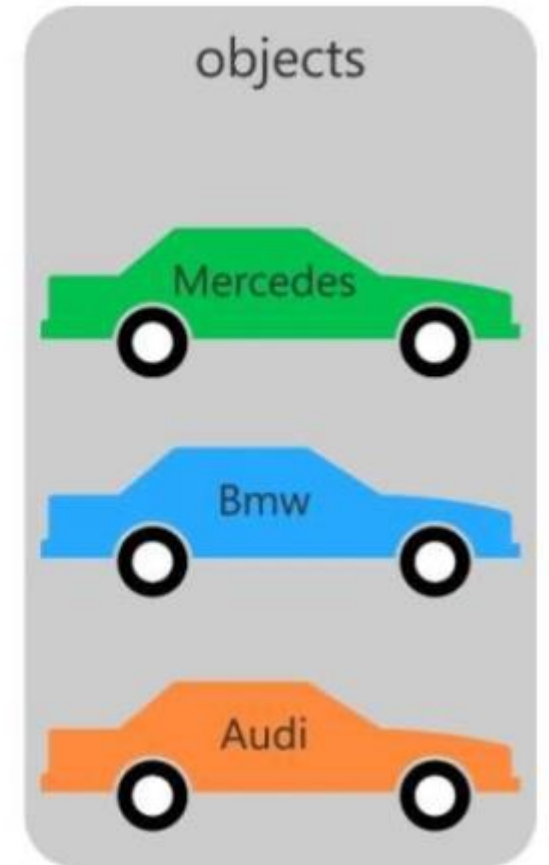
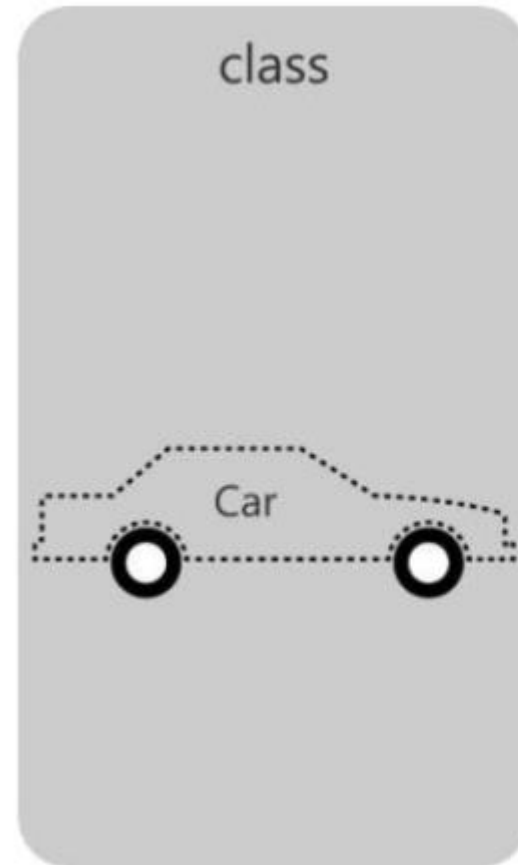
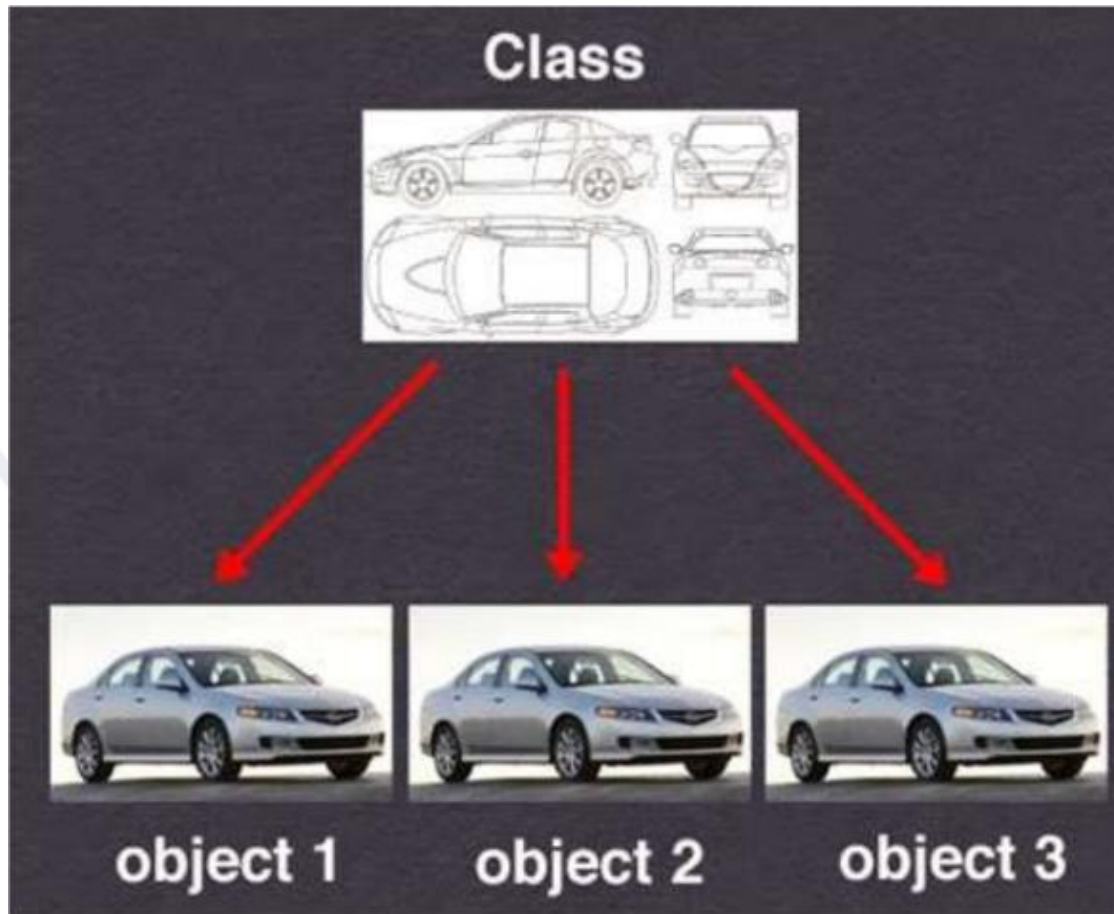
- In all programming languages, we create variable of built in data types like
`int a;`
- The above statements declare as variable a of data type int which means that a can hold any integer data.
- Similarly, we can create a variable of user defined data type like
`Student s;`
- Here student is a user defined data type and s is variable of that.
- In order to create a user, define data type java provides the concept of classes.

- Following are the points which defines a class:
 1. A class consist of data and methods.
 2. A class is a collection of objects of similar type of objects.
 3. Classes are user defined data.
 4. Once class has been defined, we can create any number of objects belonging to that class.

Class

- A class is a group of objects that has common properties. It is a template or blueprint from which objects are created
- A class in java can contain:
 1. Data members
 2. Methods
 3. Constructor
 4. Block
 5. Class and interface

Example



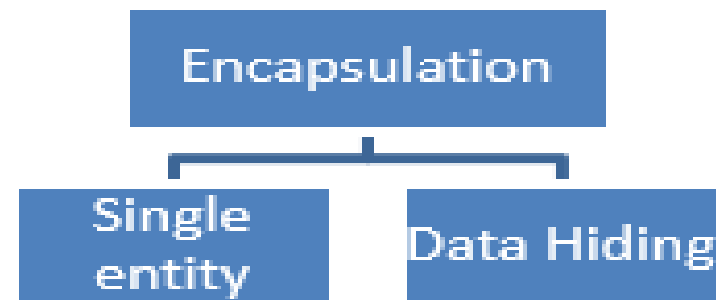
Data abstraction

- Data abstraction refers to act of representing essential features without including background details or explanations.
- Classes use the concept of abstraction. They encapsulate all essential properties of objects that they are to be created .Since the class used the concept of data abstraction they are known as abstract data types(ADT).
- Abstraction provides well defines interface and hide implementation interface



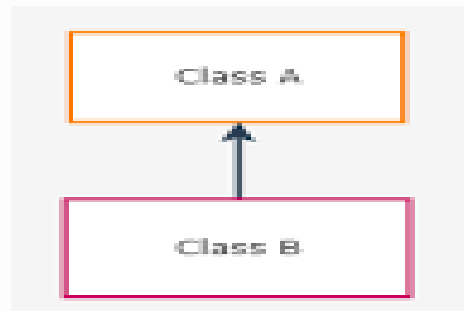
Data Encapsulation

- Data encapsulation means wrapping of data and functions into a single unit. It is most striking feature of class. The data is not accessible to the outside world and only functions which are wrapped in the class can access it.
- Thus insulation of data from direct access from program is called data hiding.



Inheritance

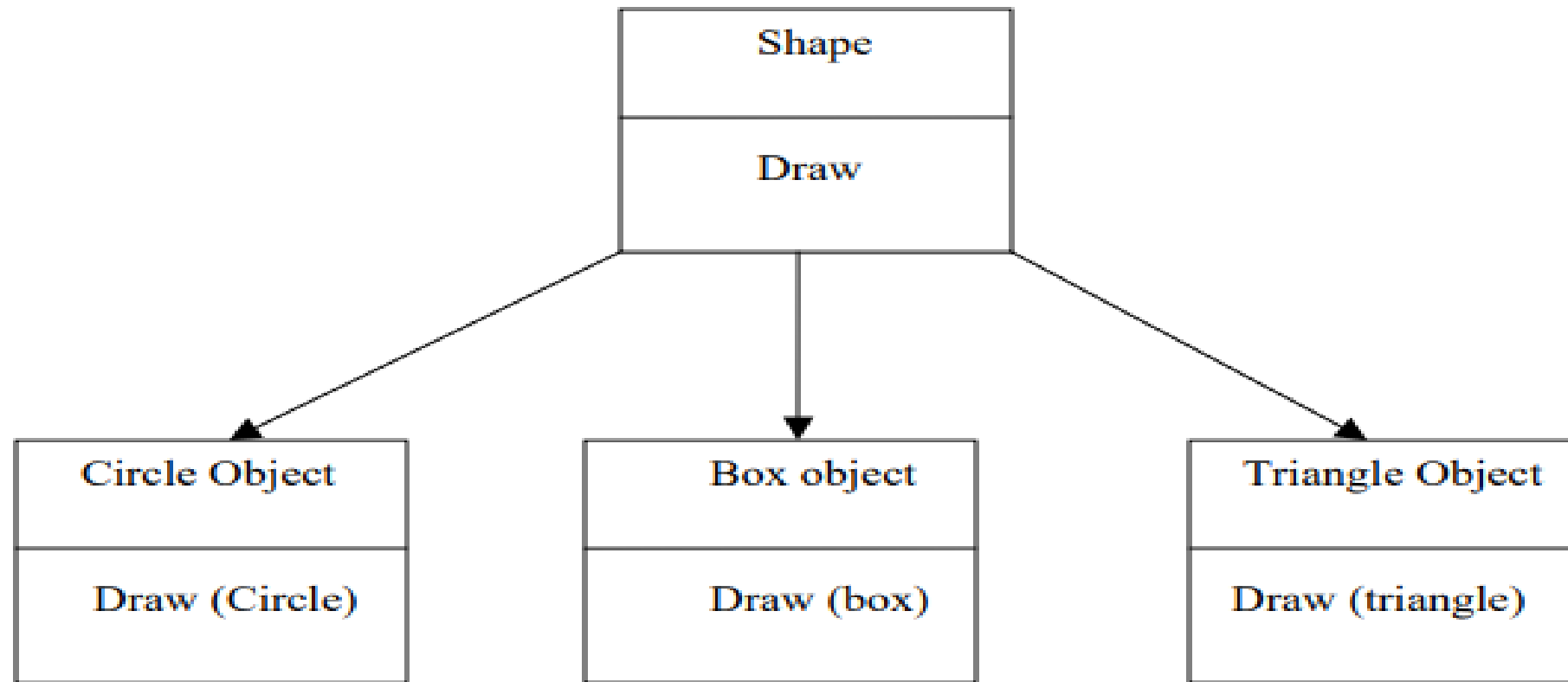
- Inheritance is the process by which object of one class can acquire the properties of the objects of other classes. Inheritance means one class of object inherits the data and behavoiur(methods) from another class .
- The old is referred as a base class and new one is called as derived class. The new class has combined features of both the classes.



Polymorphism

- Polymorphism is the ability to take more than one form.
- For eg: an operation may demonstrate different behaviour in different instance. The behavior depends upon the type of data used in the operation. As addition of two integer numbers give addition while addition of two strings give concatenation.
- illustrates that a single function name can be used to handle different number and different types of arguments. This is something similar to a particular word having several different meanings depending on the context.

Polymorphism



Dynamic Binding

- Binding refers to the linking of a procedure call to the Procedure definition (code) to be executed in response to the call.
- Binding is classified as static binding and dynamic binding.
- Static binding means that the code associated with given procedure call is known at compile time.
- Dynamic binding means that the code associated with given procedure call is not known until the time of the call at run time.
- Dynamic binding associated with polymorphism and inheritance.

Message Passing

- Message passing involves specifying the name of object, the name of the method and the information to be sent.

- Example:.

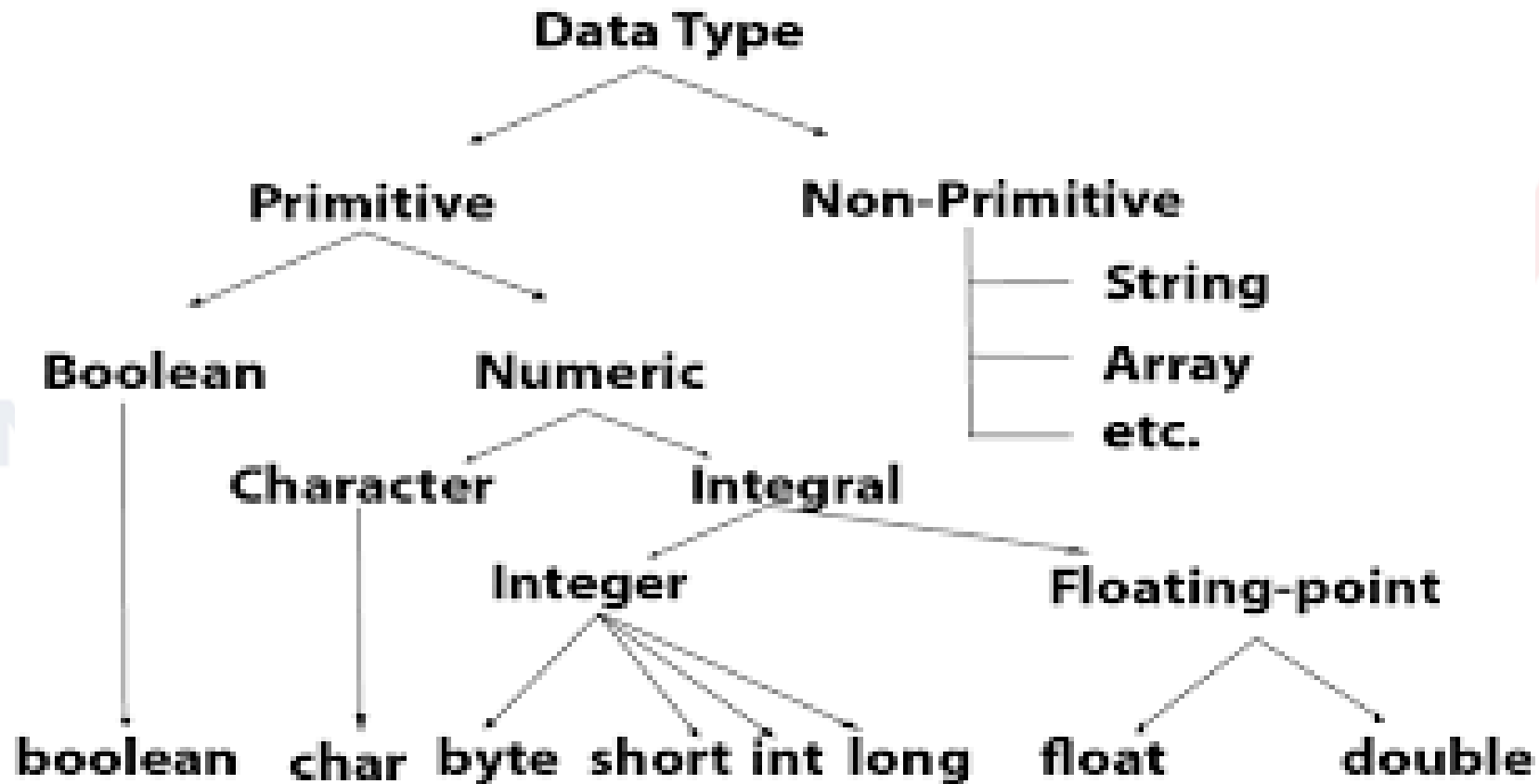
```
p.accept("abc",18);
```

- Here p is an object of Person class.
- Accept is message
- Arguments is information

Application Areas of OOP

- The promising areas of application of OOP include:
- Real-time system
- Simulation and modeling
- Object-oriented data bases
- Hypertext, Hypermedia, and expertext
- AI and expert systems
- Neural networks and parallel programming
- Decision support and office automation systems
- CIM/CAM/CAD systems

Data Types



Sr. no.	Data Type	Size in bytes
1	Boolean	1
2	char	2
3	byte	1
4	short	2
5	int	4
6	long	8
7	float	4
8	double	8

Case Study

Name (Identifier)	Student	Circle	SoccerPlayer	Car
Variables (Static attributes)	name gpa	radius color	name number xLocation yLocation	plateNumber xLocation yLocation speed
Methods (Dynamic behaviors)	getName() setGpa()	getRadius() getArea()	run() jump() kickBall()	move() park() accelerate()

Examples of classes

References

- Object Oriented Programming in java , Lecture slide by Adil Aslam
(<https://www.slideshare.net/AdilAslam4/object-oriented-programming-in-java-slide-16>)
- Programming with Java By E Balagurusamy
(https://books.google.co.in/books?id=yuGfDwAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)