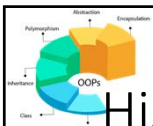


Introduction to Object Oriented Programming

Chapter 1


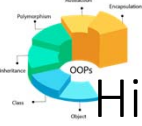
Object Oriented Programming

By DSBaral



History of Programming Languages

- The history of programming languages is a journey from basic mechanical instructions to high-level, human-friendly tools that power today's technology.
- Early history of programming language can be traced back to 1800.
- In 1801, Joseph Marie Jacquard made Jacquard Loom that used punched cards to control weaving patterns — an early example of programming instructions.
- In 1830s Charles Babbage designed the Analytical Engine, a mechanical general-purpose computer and Ada Lovelace wrote what is considered the first algorithm for a machine, making her the first programmer.


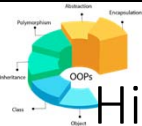


History of Programming Languages (Contd...)

- In 1940s computers used machine Language Programs that were written in binary (0s and 1s) specific to a machine's hardware.
 - It was very error-prone and hard to understand or debug.
- Later Assembly Language was developed that used mnemonics instead of binary (e.g., MOV, ADD, SUB).
 - It required an assembler to convert into machine code.
 - It was still hardware-specific, but easier than binary.

OOP, Introduction to OOP by DSBaral

3


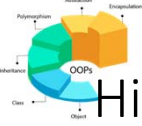


History of Programming Languages (Contd...)

- In 1950s-1960s, high-level languages (non structured) emerged in the programming arena, making programming much easier.
 - Fortran (1957): First widely used high-level language, that focused on scientific and engineering calculations. Early versions used GOTO heavily.
 - LISP (1958): Designed for artificial intelligence; based on symbolic expression and recursion. It relied heavily on recursion and lacked structured concept.
 - ALGOL 58 (1958): Introduced some structured ideas, but not widely adopted initially.
 - COBOL (1959): Focused on business data processing; also lacked structured constructs early on.
 - BASIC (1964): Easy-to-learn language; early versions relied on GOTO and line numbers. It was designed for beginners and education.

OOP, Introduction to OOP by DSBaral

4





History of Programming Languages (Contd...)

- In 1960s-1970s, structured programming started.
 - ALGOL 60 (1960): Considered the first structured language (more structured than ALGOL 58).
 - Pascal (1970): Designed explicitly for teaching structured programming.
 - C (1972): Powerful low-level structured language; widely used. Was used in writing UNIX operating system.
 - Modula-2 (1978): Successor to Pascal with modular programming features.
 - Fortran 77 (1978): Introduced structured programming features such as IF...THEN...ELSE statements and improved loop control, making programs more readable and maintainable.

OOP, Introduction to OOP by DSBaral

5





History of Programming Languages (Contd...)

- In 1980s, object-oriented programming started in programming arena.
 - C++ (1983): Extension of C with object-oriented features.
 - Objective-C (1984): Combined C and Smalltalk features, later used by Apple.
- In 1990s, the Internet and scripting started
 - Python (1991): Known for readability and simplicity.
 - Java (1995): "Write once, run anywhere" – widely used in enterprise and mobile applications.
 - JavaScript (1995): Key language for web development.
 - PHP (1995): Server-side scripting for web development.

OOP, Introduction to OOP by DSBaral



6

History of Programming Languages (Contd...)

- In recent days modern programming techniques were started with simple, performance and safety.
 - D (2001): Designed for low-level control like C++, with the simplicity, safety, and productivity.
 - C# (2000): Developed by Microsoft for .NET framework.
 - Ruby (1995, gained popularity in 2000s): Known for elegant syntax; popularized by Ruby on Rails.
 - Go (2009): Developed by Google for efficient concurrency.
 - Swift (2014): Apple's modern language for iOS/macOS development.
 - Rust (2015): Focused on safety and performance; popular in systems programming.
 - Kotlin (2011): Official Android language; interoperable with Java.
 - TypeScript (2012): Superset of JavaScript with static typing.



OOP, Introduction to OOP by DSBaral 7

Basics of Object Oriented Programming

- Objects are the entities that can be uniquely identified from others.
- In real world system everything exists in the form of objects, e.g., desk, bench, blackboard, student, teacher, car, tree are objects.
- Every object has two things, firstly its properties we call attributes and second its behavior we call function. E.g., a car object has attributes like color, number of seats, chassis number, engine number etc. and behavior like move, stop, accelerate, turn etc.
- The Object Oriented Programming is developed to model such real world system with the objective to overcome the limitation of Procedure Oriented approach.



OOP, Introduction to OOP by DSBaral 8



Procedure Oriented Programming

- In procedure oriented programming a large program is broken down into smaller manageable parts called procedures or functions and priority is given on function rather than data.
- In procedure oriented programming language, a program basically consists of sequence of instructions each of which tells the computer to do something such as reading inputs from the user, doing necessary calculation, displaying output.

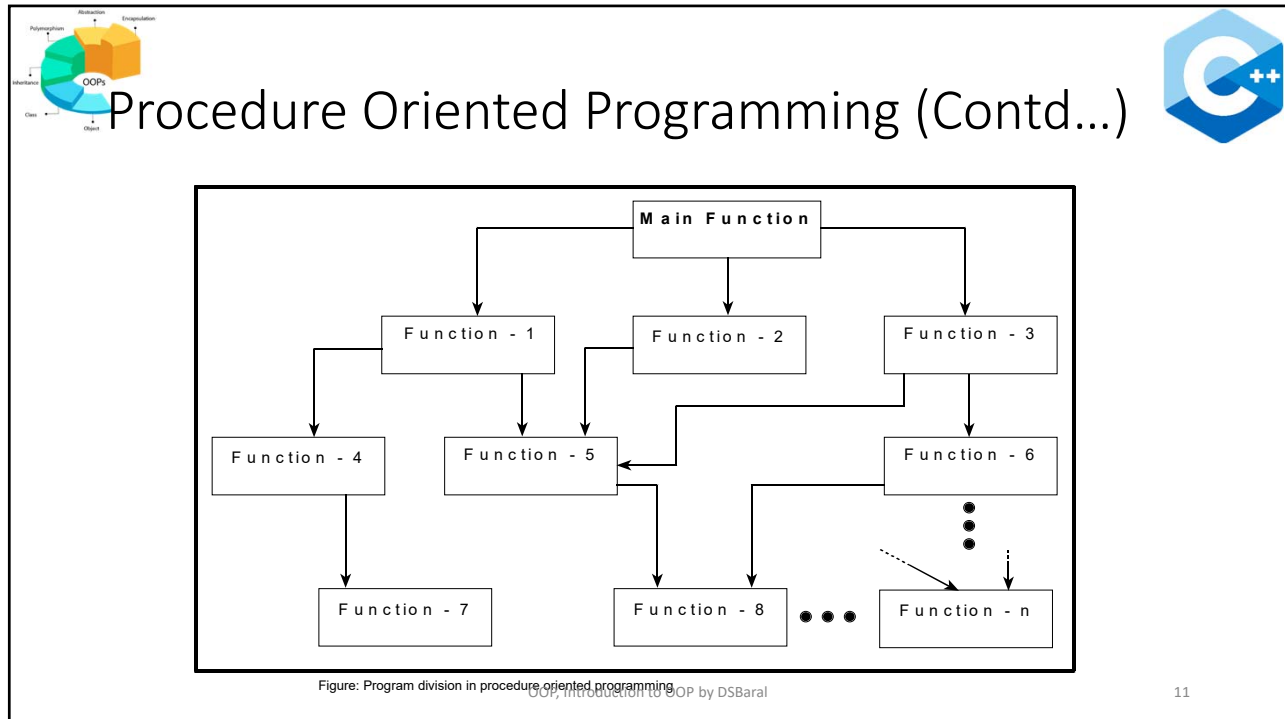
OOP, Introduction to OOP by DSBaral 9



Procedure Oriented Programming (Contd...)

- When a program becomes larger, it is then broken into smaller units called procedure or functions.
 - A number of functions are supposed to be written to accomplish such tasks.
- The primary focus of procedure oriented programming is on functions rather than data.
 - These functions do not let code duplication because of same function call from multiple locations.
 - This technique is only suitable for medium sized software applications.

OOP, Introduction to OOP by DSBaral 10





```
graph TD; Main[Main Function] --> F1[Function - 1]; Main --> F2[Function - 2]; Main --> F3[Function - 3]; F1 --> F4[Function - 4]; F1 --> F5[Function - 5]; F2 --> F5; F2 --> F6[Function - 6]; F3 --> F6; F4 --> F7[Function - 7]; F5 --> F8[Function - 8]; F6 --> Fn[Function - n];
```

Procedure Oriented Programming (Contd...)

- In procedure oriented programming two types of data local and global are used.
- Data within the function are called local data and the data which are not within any function are called global data.
- Local data are accessible to the only function where it is declared.
- Each function may access its local data as well as global data.
- The local data of one function is not accessible to other functions.

OOP, Introduction to OOP by DSBaral

12





Procedure Oriented Programming (Contd...)

- If any data is to be accessed by two or more functions it should be made global.
 - Global data are vulnerable to another programmer to be changed unknowingly.
- The separate arrangement of data and functions does a poor job of modeling things in the real world.
- High Level Programming Languages like COBOL, FORTRAN, Pascal, C are common procedure oriented programming languages

OOP, Introduction to OOP by DSBaral

13



Procedure Oriented Programming

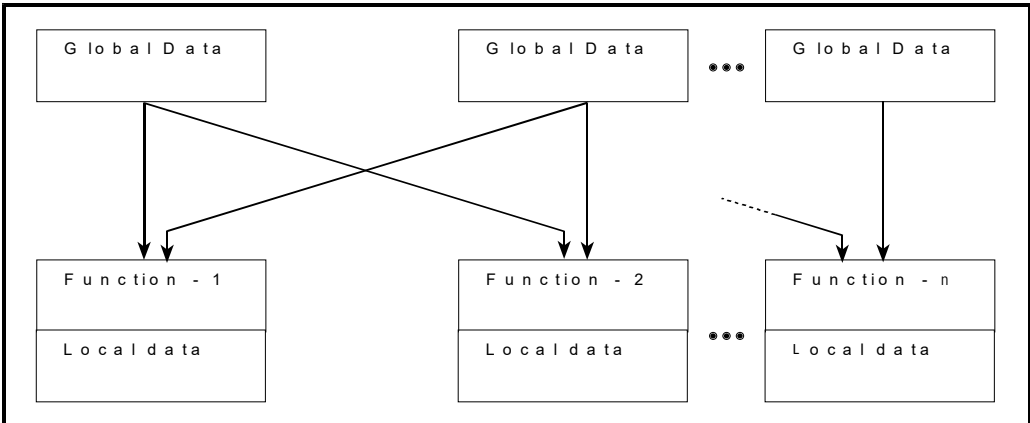




Figure: Usage of global data in procedural program

OOP, Introduction to OOP by DSBaral



14



Characteristics of POP

- A large program is broken down into small manageable procedures or functions.
- Procedure oriented programming focuses on procedure or function rather than data.
- For sharing a common data among different functions the data is made global.
 - Global data are vulnerable, may be altered by the function.
- The program design of procedure oriented programming follows top down methodology.



OOP, Introduction to OOP by DSBaral 15



Limitations of POP

- Focus is on functions rather than data.
- In large program, it is difficult to identify belonging of global data.
 - The use of global data is error prone and it could be an obstacle in code maintenance and enhancements as the modification of global data requires the modification of those functions using it.
- Maintaining and enhancing program code is difficult.
- It does not model real world problem very well because functions are action oriented and do not really correspond to the elements of problem.

OOP, Introduction to OOP by DSBaral 16





Object Oriented Programming

- The limitations of the procedure oriented programming approach are the motivating factor in the invention of objected oriented approach.
- In OOP, data are treated as a critical element in the program and restricts freely transformation of data around the system
- Data are associated with functions that operate on it and protect it from accidental modification outside functions.
- OOP approach permits decomposition of a problem into entities called objects and then build data and function around them.

OOP, Introduction to OOP by DSBaral

17

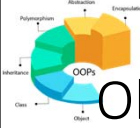


Object Oriented Programming (Contd...)


- Data of an object are accessible only by the function belonging with the object but function of one object may access the function of another object.
- Object oriented programming is a programming methodology that associates data structures with a set of operators which act upon it.
- In OOP, an instance of such an entity is known as object.
- The OOP is a method of implementation in which programs are organized as cooperative collections of objects, each of which represents an instance of some class and whose classes are all members of a hierarchy of classes united through the property called inheritance.

OOP, Introduction to OOP by DSBaral

18



Object Oriented Programming (Contd...)



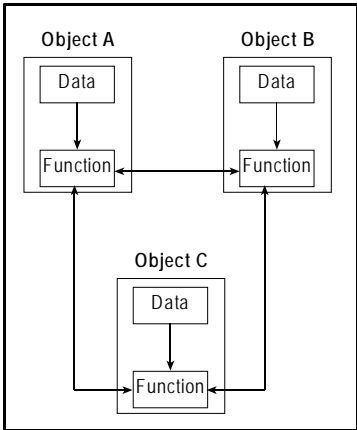




Figure: Object Oriented Approach

OOP, Introduction to OOP by DSBaral

19




Characteristics of OOP




- Emphasis is on data rather than procedures.
- Programs are divided into objects.
- Function and data are tied together in a single unit.
- Data is made hidden to prevent from accidental alteration from other function or objects.
- Data access is done through the visible functions so that communication between objects is possible.
- Data structures are modeled as objects.
- Follows Bottom up approach of program design methodology.

OOP, Introduction to OOP by DSBaral

20




POP Vs OOP




Procedure Oriented Programming	Object Oriented Programming
Emphasis is given on procedures.	Emphasis is given on data.
Programs are divided into functions.	Programs are divided into objects.
Follow top-down approach of program design.	Follow bottom-up approach of program design.
Generally data cannot be hidden.	Data can be hidden, so that non-member function cannot access them.

OOP, Introduction to OOP by DSBaral

21





POP Vs OOP (Contd...)



Procedure Oriented Programming	Object Oriented Programming
It does not model the real world problem perfectly.	It models the real world problem very well.
Data move from function to function.	Data and function are tied together. Only related function can access them
Maintaining and enhancing code is still difficult.	Maintaining and enhancing code is easy.
Code reusability is not efficient.	Code reusability is more efficient in compare to procedure oriented approach.
Examples: FORTRAN, COBOL, Pascal, C	Example: C++, JAVA, Smalltalk

OOP, Introduction to OOP by DSBaral

22


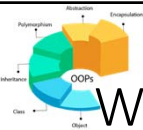


Why is OOP Popular?

- Following are the reasons why OOP has become popular
 - The hope that it will quickly and easily lead to increased productivity and improved reliability (help solve the software crises).
 - The desire for an easy transition from existing languages.
 - The resonant similarity to techniques of thinking about problems in other domains.

OOP, Introduction to OOP by DSBaral

23





Why is OOP Popular? (Contd...)

- Object-Oriented programming is just the latest in a long series of solutions that have been proposed to help solve the “software crises”.
 - The software crises simply means that our imaginations, and the tasks we would like to solve with the help of computers, almost always nearly surpass our abilities.
- Object-oriented techniques facilitate the creation of complex software systems but is not a magical weapon.
- Programming a computer is still one of the most difficult tasks ever undertaken by humankind; becoming proficient in programming requires talent, creativity, intelligence, logic, the ability to build and use abstractions, and experience even when the best of tools are available.

OOP, Introduction to OOP by DSBaral



24



Why is OOP Popular? (Contd...)

- Managers and programmers may think that a C or Pascal programmer can be changed into a C++ or Object Pascal programmer with no more effort than the addition of two characters to the programmer's job title.
- However, this is not true because Object-Oriented programming is a new way of thinking about what it means to compute, about how we can structure information inside a computer.
- To become proficient in object-oriented techniques requires a complete reevaluation of traditional software development techniques.



OOP, Introduction to OOP by DSBaral 25



Language and Thought

- The 'real world' is unconsciously built up on the language habits of the group and the language habits of our community influence certain choices of interpretation. (Sapir-Whorf hypothesis)
 - thoughts in one languages may not be translated in other languages
- The languages we speak influence directly the way in which we view the world.
 - This is true not only for natural languages but also for artificial languages such as those we use in programming computers.
- Eskimos languages have many words to describe various types of snow such as wet, fluffy, heavy, icy, and so on.
- A different language can lead one to view the world in a different fashion (with English we cannot describe snow)



OOP, Introduction to OOP by DSBaral 26

Language and Thought (Contd...)

- To make effective use of object oriented principles requires one to view the world in a new fashion.
- Simply using an object oriented language (such as C++) does not, by itself, force one to become an object oriented programmer but using one will simplify the development of object oriented solutions.
- The language in which a programmer thinks a problem will be solved will influence the way in which an algorithm is developed.
- According to Sapir-Whorf hypothesis, thoughts in one languages may not be translated in other languages.
- Any computation for which there exists an effective procedure can be realized by a Turing machine (Church's Conjecture)



OOP, Introduction to OOP by DSBaral 27

Language and Thought (Contd...)

- If we accept Church's conjecture, then any language in which it is possible to simulate a Turing machine is sufficiently powerful to perform any realizable algorithm.
 - Church's conjecture is almost the exact opposite of the Sapir-Whorf hypothesis.
 - Church's conjecture states that in a fundamental fashion all programming languages are identical.
- Any idea that can be expressed in one language can, in theory, be expressed in any language.
- Many linguists reject the Sapir-Whorf hypothesis and instead adopt a sort of "Turing equivalence" for natural languages.



OOP, Introduction to OOP by DSBaral 28

Language and Thought (Contd...)

- Like the people from hotter climate cannot describe snow object-oriented techniques do not provide any new computational power which permits problems to be solved that cannot, in theory be solved by other means.
 - But object oriented techniques do make it easier and more natural to address problems in a fashion that tends to favor the management of large software projects.
- Thus, for both computer and natural languages it is the case that the language will direct thoughts, but cannot prohibit thoughts.



OOP, Introduction to OOP by DSBaral 29

OOP A New Paradigm

- Object-oriented programming is often referred to as a new programming paradigm.
 - Other programming paradigms sometimes mentioned include the imperative-programming paradigm (languages such as Pascal or C), the logic programming paradigm (Prolog), and the functional programming paradigm (FP or Haskell).
- A programming paradigm is a way of conceptualizing what it means to perform computation, and how tasks that are to be carried out on a computer should be structured and organized.
- The style of problem solving embodied in the object-oriented technique is frequently the method used to address problems in everyday life.
 - Thus, computer novices are often able to grasp the basic ideas of object-oriented programming easily, whereas those people who are more computer literate are often blocked by their own preconceptions.

OOP, Introduction to OOP by DSBaral 30






A Way of Viewing the World

- Let us see a real-world situation, and then examine how we could make the computer more closely model the techniques employed.
- Suppose grandson wish to send some flowers to his grandmother, who lives very far away, for her birthday.
- Even sending flower is not difficult, he can just go down to the local florist, describe the kinds and numbers of flowers he want to send, and his grandmother's address, and he can be assured that the flowers will be delivered conveniently and automatically.

OOP, Introduction to OOP by DSBaral

31






Agents, Responsibility, Messages and Methods

- To solve the problem the boy choose to find an appropriate **agent**, local florist, and to pass to her a message containing my request.
- It is the **responsibility** of the florist to satisfy the grandson's request.
- There is some method, that is, some algorithm or set of operations, used by the florist (agent) to do this.
- The boy do not need to know the particular method florist will use to satisfy his request and he do not need to know the details.
 - This information is usually hidden from his inspection.

OOP, Introduction to OOP by DSBaral



32

Agents, Responsibility, Messages and Methods (Contd...)

- Florist might also delivers a slightly different message to another florist in grandmother's city to makes the arrangement and passes it, along with yet another message, to a delivery person, and so on, with a sequence of requests from one agent to another.
- Action is initiated in object-oriented programming by the transmission of a *message* to an *agent* (an *object*) responsible for the action.
- The message encodes the request for an action, and is accompanied by any additional information (arguments) needed to carry out the request.
- The *receiver* is the agent to whom the message is sent and if it accepts the message, it accepts the *responsibility* to carry out the indicated action.
- In response to a message, the receiver will perform some *method* to satisfy the request



OOP, Introduction to OOP by DSBaral 33

Agents, Responsibility, Messages and Methods (Contd...)

- The use of agent is like using the ready made reusable components do have the work done instead of writing everything by the programmer himself/herself.
- One of the important part of object oriented programming is the development of reusable components and use them when needed.
- There is a distinction between message passing and procedure calling
 - In message passing, there is a designated receiver, and the interpretation, that is, the selection of a method to execute in response to the message and may vary with different receivers. (like response to a flower message by a dentist and florist)
 - In procedure call there is no distinct receiver (assumption can be made that first argument is the receiver)


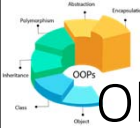
OOP, Introduction to OOP by DSBaral 34



Concept of Object Oriented Programming

- Object/Class
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Reusability
- Dynamic binding
- Message passing

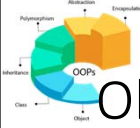
OOP, Introduction to OOP by DSBaral 35




Object

- Object are entities with properties (data member and member functions)
- Objects can be a person, a place, a bank account, or any item that the program must handle. For example, Automobiles are objects as they have size, weight, color etc. as attributes (that is data) and starting, pressing the brake, turning the wheel, pressing accelerator pedal etc. as operation (those are functions).
- Specific automobile has its specific size, weight, color etc. but all the automobiles have common operations

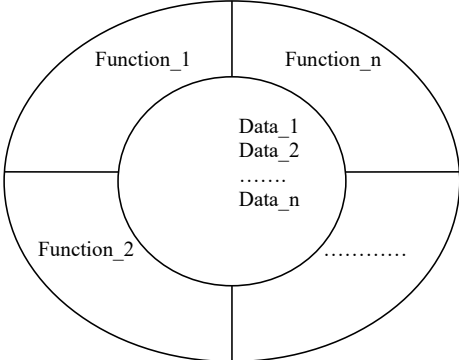
OOP, Introduction to OOP by DSBaral 36



Object (Contd...)



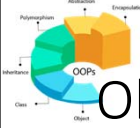
Object Name
Data Data_1 Data_2 Data_n
Functions Function_1 () Function_2 () Function n ()




Model of General Object

OOP, Introduction to OOP by DSBaral

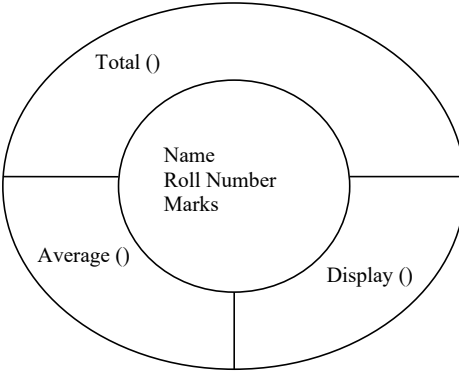
37



Object (Contd...)



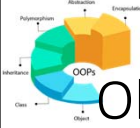

Object: Student
Data Name Roll number Marks
Functions Total () Average () Display ()



Model of Student as object

OOP, Introduction to OOP by DSBaral

38

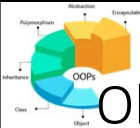



Objects (Contd...)

- Objects in different scenario can be
- Physical Objects
 - Bus in Traffic System
 - Atom in chemical composition
 - Diode in electronic system
 - Leader in political system
- Graphical User Interface
 - Menu
 - Button
 - Toolbar
 - Combo box
 - Text box

OOP, Introduction to OOP by DSBaral

39





Objects (Contd...)

- Data Structure
 - Vector
 - Stack
 - Queue
 - Tree
- Human Disciplines
 - Actor
 - Singer
 - Musician
 - Student
 - Teacher

OOP, Introduction to OOP by DSBaral

40





Objects (Contd...)

- Geometrical Shapes
 - Point
 - Line
 - Triangle
 - Circle
- User defined data
 - Distance
 - Currency
 - Time
 - Date
 - Complex Number

OOP, Introduction to OOP by DSBaral

41





Class

- Classes specify the entire category/group of objects
- An object specify a single entity/item while the class specify all objects of same type.
- A specific red car of 800kg weight is a specific object and group of cars with property color and weight is a class.
- Under that class there can be any number of cars with different color and weight.
- All objects are instances of a class.
- The method invoked by an object in response to a message is determined by the class of the receiver.
- All objects of a given class use the same method in response to similar messages

OOP, Introduction to OOP by DSBaral



42

Class (Contd...)

- The entire construct of objects can be represented by a user defined data type in programming.
- The class is the user defined data type (like `struct` in C) used to declare the objects.
- Objects are the variable of the user defined data type implemented as class.
- Once a class is defined, we can create any number of objects of its type, that is, class is the data type of that object.
- Class specifies what data and functions will be included in objects of that class.
- A class is a template that specifies data and their operations

OOP, Introduction to OOP by DSBaral 43






Class (Contd...)

Class: Student
Data Members Name Registration_number Marks
Function Members Tot_marks () Percentage_marks() Decide_division ()

- One of the objects of student can have following values
 - Name = "Bishal"
 - Registration_number = 200876255
 - Marks = {66, 77, 51, 48, 82}
- The functions will do the task as accordingly.
- Each object of the class is an instance of its class with its own value for the attribute that shares the attribute name and operations with other instances of the class



OOP, Introduction to OOP by DSBaral 44



Abstraction

- Abstraction is the concept of object oriented programming that “shows” only essential attributes and “hides” unnecessary information.
- Abstraction feature of OOP hides the internal details of how any object does its work.
- It only provides the interface (public functions) to use the service (internal implementation) that the object provides.
- It focuses the outside view of an object, separating its essential behavior from its implementation.



OOP, Introduction to OOP by DSBaral 45



Abstraction (Contd...)

- For example a car driver can drive the car without knowing the details of the car.
- You can order coffee from the waiter and have it without knowing how it is made
- The class is a construct in object oriented programming for creating user-defined data for abstraction.
- The mathematical model of a data structure and its operations presented together is called the abstract data type (ADT). For example, list, queue, stack etc.



OOP, Introduction to OOP by DSBaral 46



Encapsulation

- The mechanism of combining data and function together into a single unit is called encapsulation
- We can assume encapsulation as a protective wrapper that prevents the data being accessed by other code defined outside the wrapper.
- By making use of encapsulation we can easily achieve abstraction.
- The class feature in OOP provides the encapsulation that also hides the complexity.



OOP, Introduction to OOP by DSBaral 47



Encapsulation (Contd...)

- The public (visible) interface of a class provides information for the external user of the class about its data and functions
- The private functions and data can only be accessed by code inside the class.
- This insulation of data from direct access by the program is called data hiding which prevents data from accidental alteration.



OOP, Introduction to OOP by DSBaral 48



Inheritance

- Inheritance is the process of creating new classes based on the existing class.
- The new class acquires features from the existing class and adds more features in it.
- This is possible by deriving a new class (derived class) from the existing one (base class).
- This process of deriving a new class from the existing class is called inheritance.
- Inheritance models the natural hierarchical classification of real world system perfectly.
- It allows the extension and reuse of existing code without having to rewrite the existing code.



OOP, Introduction to OOP by DSBaral 49



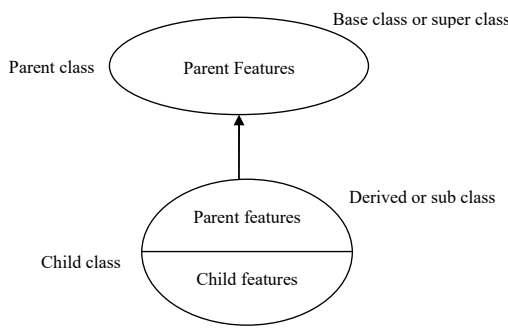
Inheritance (Contd...)

- We naturally view that the whole world is made up of objects.
- Many objects are related to each other in a hierarchical way, such as vehicle, four-wheeler, and car.
 - If we describe vehicle in an abstract way, the attributes may be such as color, weight etc.
 - All vehicles have common behavioral aspect like; they move, accelerate, turn and stop.
 - Four wheeler is more specific case of a vehicle with more specific attributes such as number of seats, engine number etc.
 - The class that represent vehicle is a base class (or super class) and class four-wheeler is a derived class (or subclass)

OOP, Introduction to OOP by DSBaral 50

Inheritance (Contd...)



Base class or super class

Parent class

Derived or sub class

Child class

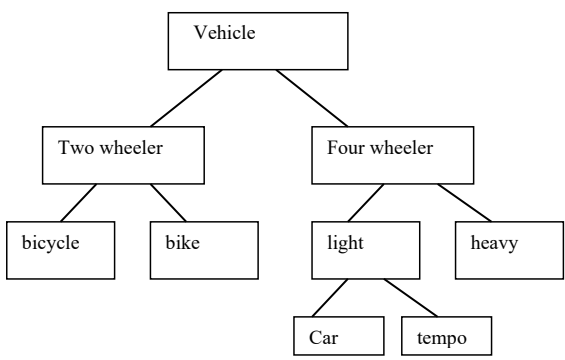




Figure: A child class inherited from parent class

Figure: Example of inheritance class hierarchy

OOP, Introduction to OOP by DSBaral

51






Polymorphism

- Polymorphism means “having many forms”.
- The polymorphism allows different objects to respond to the same operation in different ways, the response being specific to the type of object.
- The different ways of using same function or operator depending on what they are operating on is called polymorphism.
- Example of polymorphism in OOP is operator overloading, function overloading.
 - Still another type of polymorphism exist which is achieved at run time also called dynamic binding.

OOP, Introduction to OOP by DSBaral

52





Polymorphism (Contd...)

- For example operator symbol '+' is used for arithmetic operation between two numbers, however by overloading (means given additional job) it can be used with our objects to represent complex number or date and time.
- The same operator '+' can also be used for other purpose apart from mathematical operation such as string concatenation
- When same function name is used in defining different function to operate on different data (type or number of data) then this feature of polymorphism is function overloading.

OOP, Introduction to OOP by DSBaral

53





Reusability

- Like library functions in procedural programming a class in Object Oriented Programming can be distributed for further use.
- In OOP, the concept of inheritance provides the idea of reusability.
- Once a class is completed and tested, it can be distributed for the development of other programs too.
- The programmer can add new features or make some changes or can derive new classes from the existing class.
- For example if we have Employee class which is tested and verified then we can create other classes Manager, Clerk, Cashier, Assistant etc. from that class without writing existing code.

OOP, Introduction to OOP by DSBaral

54





Reusability (Contd...)

- This inheritance example is another example where a class models the real world.
- If a software company creates generic classes for one project then the company can use the same class and its extensions in the new project with less time, effort and investment.
- The features functions, classes, inheritance and templates provides the reusability feature.
- The code reusability features help us in writing less redundant code.

OOP, Introduction to OOP by DSBaral

55





Dynamic Binding

- The process of selecting (binding) a function during runtime in response to the call is called dynamic binding.
- There are two types of binding one is static binding (also called early binding) and another is dynamic binding (also called late binding).
- Function overloading and operator overloading construct in OOP are the examples of early binding.
- The early binding occurs at the compile time, so, it is called compile time polymorphism.

OOP, Introduction to OOP by DSBaral

56





Dynamic Binding (Contd...)

- Dynamic binding means that the code associated with a given function call is not known until the run time, so, it is called as run time polymorphism.
- Dynamic binding is possible only when we use inheritance and access the objects through pointers.
- If classes Circle, Box, and Triangle are derived from same class Shape and all of the classes define a function draw(), then during the function call draw() through the pointer variable of Shape class an appropriate function belonging to that class is invoked.

OOP, Introduction to OOP by DSBaral

57



Dynamic Binding (Contd...)

Class Shape
draw()

Circle Object
draw(circle)



Box Object
draw(box)

Triangle Object
draw(Triangle)

Figure: Dynamic binding for function draw()

OOP, Introduction to OOP by DSBaral



58

Message Passing

- Object oriented language have message driven communication, that is, a message is sent to an object.
 - Communications among the objects are analogous to exchanging messages among people.
 - Object communicates with each other by sending and receiving message (information).
- The communication in object oriented programming is between the objects themselves instead of functions in procedural languages.
- A message for an object is a request for execution of a method (function) and therefore will invoke a function or procedure in receiving object that generate the desired result.

OOP, Introduction to OOP by DSBaral 59

Message Passing (Contd...)



- Message passing involves specifying the name of the object, name of the function (message) and the information (arguments to function) to be sent.
 - The message for an object is a request for the execution of a function belonging to an object which generates the desired result for the given argument.

```

      student.fee (name) ;
      ^       ^       ^
      |       |       |
    object  message  information
  
```

- Communication between the objects takes place as long as their existence.
- Objects are created and destroyed automatically whenever needed.
- In above example student is regarded as an object sending the message fee to find the fee to be paid by the student with the give fee category name.



OOP, Introduction to OOP by DSBaral 60



Advantages of OOP

- The main advantages of OOP are:
 - Redundant code is eliminated by various techniques like class, inheritance and templates.
 - Through data hiding, programmer can build secure programs.
 - Existing classes can serve as library class for further enhancements.
 - Division of program into objects makes software development easy.
 - Software complexity is less severe than conventional programming techniques.
 - Because of dynamic binding, addition of new classes of objects at run time is possible without modifying the existing code.


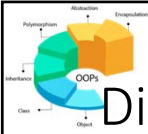
OOP, Introduction to OOP by DSBaral 61



Advantages of OOP (Contd...)

- The limitation realized in base class can be fulfilled in derived class without writing even a single piece of code in the base class.
- Upgrading and maintenance of software is easily manageable.
- System can be easily upgraded from small to large systems.
- Message passing technique makes the interface simpler with external systems.
- Models real world system perfectly.
- Code reusability is much efficient than conventional programming.

OOP, Introduction to OOP by DSBaral 62


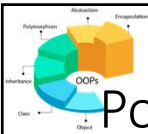


Disadvantages of OOP

- Some of the disadvantages of OOP are
 - Compiler and runtime overhead.
 - Re-orientation of software developer to object-oriented thinking.
 - Requires the mastery in software engineering and programming methodology.
 - Benefits only in long run while managing large software projects.
 - The message passing between many objects in a complex application can be difficult to trace & debug.

OOP, Introduction to OOP by DSBaral

63



Popular Object Oriented Languages

- Some of the popular object oriented languages are
 - C++
 - Simula
 - Smalltalk
 - Eiffel
 - Java
 - C#
 - D
 - Python
 - PHP

OOP, Introduction to OOP by DSBaral

64