
Object-Oriented Programming
Lesson 01: Introduction to Object-Oriented Technology
Capgemini

Lesson Objectives

- In this lesson, you will learn:
  - What is Object-Oriented Programming?
  - Why Object-Oriented Programming?
    - Object-Oriented Programming versus traditional software development methodologies
    - Benefits of Object-Oriented technology

1.1: Object Oriented Concepts

Example: Scenario from Banking System

- Geetha and Mahesh hold accounts in Bank XYZ Ltd. Geetha has a savings as well as a current account with the bank. Mahesh only has a current account. As customers of the bank, Geetha and Mahesh can deposit or withdraw money from their accounts as per the norms and policies defined by the bank on savings and current accounts.

- Bank XYZ Ltd. continuously adds new customers to its existing customer base. Of course, some its customers may also want to close their accounts due to changing needs of the customer.

If you consider this scenario for a Banking System, what services and features do you expect the bank to offer? Who are the customers mentioned for this bank? What operations can they perform on their accounts?

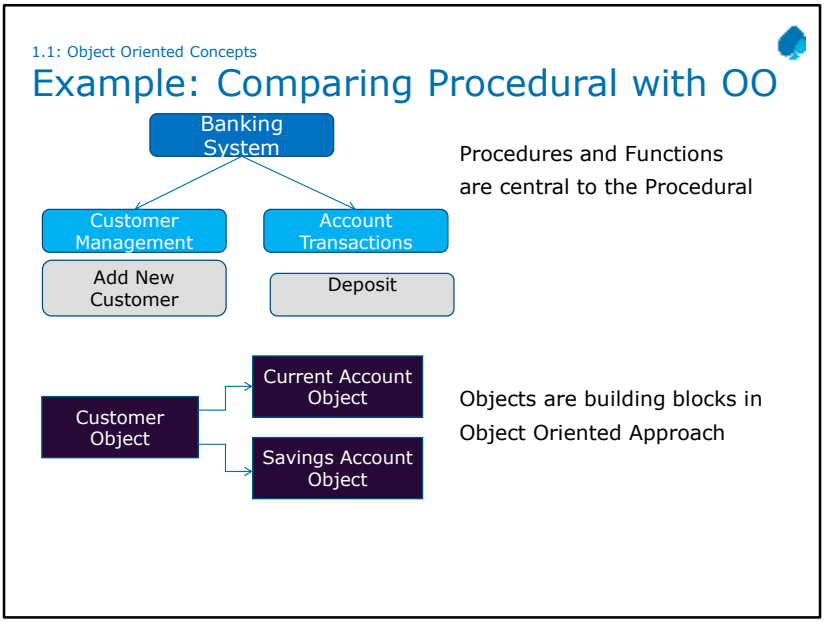
1.1: Object Oriented Concepts

What is Object-Oriented Programming

- OOP is a paradigm of application development where programs are built around objects and their interactions with each other.
  - An Object Oriented program can be viewed as a collection of co-operating objects.

Can you think of a collection of co-operating objects in the scenario from Banking System?

What is Object-Oriented Programming?
The object oriented approach is a fundamental shift from the procedural approach. Instead of functions and procedures being central to the program, in the OO world, we have objects that are the building blocks. An OO program is made up of several objects that interact with each other to make up the application.
For example: In a Banking System, there would be Customer objects pertaining to each customer. Each customer object would own its set of Account Objects, pertaining to the set of Savings and Current Accounts that the customer holds in the bank.
Today, most programming languages are object oriented.
For example: Java, C++, C#
Why do you think most of today's programming languages are object oriented? Are there any advantages of OO languages?



Consider the Banking System. In the procedural approach, we would try to find the top level modules of the application. Eg. A Module to maintain customers, another to maintain accounts and so on. In each of these modules, we would have procedures and functions to take care of different features. Eg. Procedure/Function to add customer or delete customer in the module for maintaining customer. Or procedures/ functions to deposit and withdraw in the module for maintaining accounts. So in the procedural approach, we identify modules, then identify procedures/functions – this is like a top down approach to system development.
Procedures and functions are the building blocks of the application in the procedural approach.
However, in the OO approach, it is the objects which are the building blocks. If we reconsider the same system, we would have objects for customer “Geetha” and customer “Mahesh”; we would also have objects for their respective savings and current accounts. These objects would interact with each other for us to achieve the desired features of the application.

1.1: Object Oriented Concepts

Why Object-Oriented Programming

- There are problems associated with structured language, namely:
  - Emphasis is on doing things rather than on data
  - Most of the functions share global data which lead to their unauthorized access
  - More development time is required
  - Less reusability
  - Repetitive coding and debugging
  - Does not model real world well

Why Object-Oriented Programming?

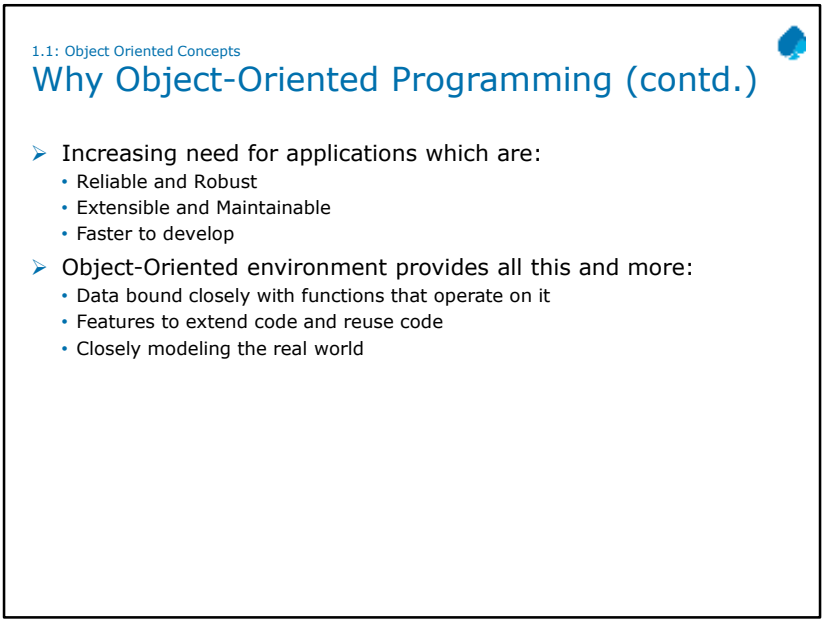
Before the advent of Object-Oriented technology, the primary software engineering methodology was structured or procedural programming. Some drawbacks of this approach are as follows:

Structured programming is based around data structures and subroutines. Data structures are simply containers for the information needed by subroutines. Thus, emphasis is almost entirely on algorithm required to solve a problem.

Data is openly accessible to other parts of the program, which is risky.

Structured programming tends to produce a design that is unique to that problem (thus non-reusable). Reusing code from another project usually involves a lot of effort and time. Moreover, since the emphasis is on functionality, functionality change might force entire code to be modified, thus increasing development time.

In structured programming, while analysis starts with a consideration of real-world problems, the real-world focus is lost as requirements are transformed into a series of data flow diagrams.



Why Object-Oriented Programming? (contd.)
With increasing complexity of software applications, some of the “must have” features are reliability, robustness, and maintainability. With increasing competition, high productivity which aids faster turn around times for application development and deployment is key.
Object-Oriented programs offer features which allow meeting the above goals. Binding of data and functions together means that data cannot be accessed unless designed for it. There is no possibility of mistakenly corrupting data. Decomposition in terms of objects allow for easily building new programs using existing objects and adding features to existing objects.
Moreover, the Object-Oriented world very closely models the real world, making it much more intuitive and faster to develop. The objects themselves often correspond to phenomena in the real world that the system is going to handle.
For example: An object can be an invoice in a business system or an employee in a payroll system. Thus, OO is natural way programming, i.e., “real life” objects are mapped as is in “programming” as classes / objects.

1.1: Object Oriented Concepts

What is Object-Oriented Programming

- Some of the major advantages of OOP are listed below:
  - Simplicity
  - Modularity
  - Modifiability
  - Extensibility
  - Maintainability
  - Re-usability

Advantages of OOP:

Simplicity: Software objects model the real world objects. Hence the complexity is reduced and the program structure is very clear.
Modularity: Each object forms a “separate entity” whose internal workings are decoupled from other parts of the system.
Modifiability: It is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only “public interface” that the external world has to a class is through the use of “methods”.
Extensibility: Adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones.
Maintainability: Objects can be separately maintained, thus making locating and fixing problems easier.
Re-usability: Objects can be reused in different programs.

OOP is more than just learning a new language. It requires “a new way of thinking”. The idea is primarily not to concentrate on the cornerstones of procedural languages - data structures and algorithms, instead think in terms of “objects”.

1.1: Object Oriented Concepts

Features of OOP

- OO Technology is based on the concept of building applications and programs from a collection of "reusable entities" called "objects".
  - Each object is capable of receiving and processing data, and further sending it to other objects.
  - Objects represent real-world business entities, either physical, conceptual, or software.
    - For example: a person, place, thing, event, concept, screen, or report

Features of OOP:

Models built by using Object-Oriented technology can be smoothly implemented in any software, by using "Object-Oriented modeling language". These models also easily adjust to changing requirements.

It is based on best practices. As a result, the systems developed by using Object-Oriented technology are stable with a baselined architecture. The systems are more reliable, scalable, and succinct. They are more easily maintained and adaptable to change.

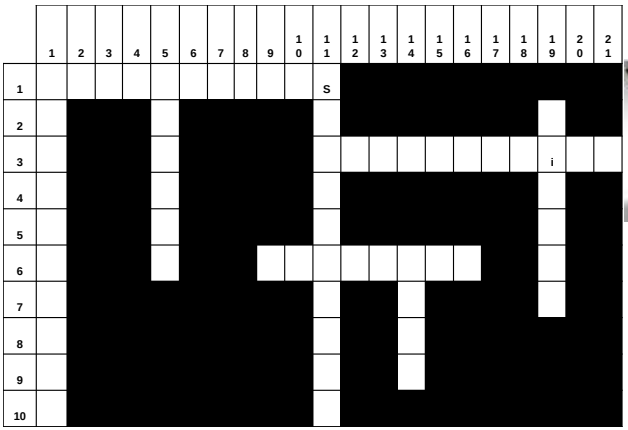
Summary

- In this lesson, you have learnt:
  - The Object-Oriented Programming approach for software development
  - How Object-Oriented technology is used to design and develop stable and dynamic systems
  - Advantages of Object-Oriented Programming

Review Question

- Question 1: Which of the following are features of Structured programming:
  - Option 1: Based on Data structure
  - Option 2: Emphasis on data
  - Option 3: Reusability
  - Option 4: Produces a design unique to the problem
- Question 2: Objects can be grouped together in different ways to form new programs.
  - True / False
- Question 3 : ___ methodology specifies the steps to be sequentially followed by a computer to execute a program.

Review Question: Crossword



Clues.: 10 rows (1-10) , 21 Cols (1-21), (Row,Col) Combination

Across:

1-1: Structured programming is based around this (11)

3-11: One of the benefits of Object-Oriented Programming (11)

6-9: Existing objects can be _____ (8)

Below:

1-1: One of the major advantages of OOP (10)

1-5: Data and functions are build around this (6)

1-11: This program data is openly accessible to other parts of the program (10)

2-19: This helps in securing the data (6)

6-14: OOP places emphasis on _____ rather than algorithm (4)