qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| **Object Oriented Programming**  Shahzad Ali Rana  Lecturer GCUF  03006641562  /shahzad.rana.127 |

**Evolution Of Object Oriented Paradigm**

**A Brief History**

The object-oriented paradigm (pattern or model) took its shape from the initial concept of a new programming approach, while the interest in design and analysis methods came much later.

* The first object-oriented language was Simula (Simulation of real systems) that was developed in 1960 by researchers at the Norwegian Computing Center.
* In 1970, Alan Kay and his research group at Xerox PARK created a personal computer named Dynabook and the first pure object-oriented programming language (OOPL) - Smalltalk, for programming the Dynabook.
* In the 1980s, Grady Booch published a paper titled Object Oriented Design that mainly presented a design for the programming language, Ada. In the ensuing editions, he extended his ideas to a complete object-oriented design method.
* In the 1990s, Coad incorporated behavioral ideas to object-oriented methods.

The other significant innovations were Object Modeling Techniques (OMT) by James Rumbaugh and Object-Oriented Software Engineering (OOSE) by Ivar Jacobson.

**Object-Oriented Analysis**

Object-Oriented Analysis (OOA) is the procedure of identifying software engineering requirements and developing software specifications in terms of a software system’s object model, which comprises of interacting objects.

The main difference between object-oriented analysis and other forms of analysis is that in an object-oriented approach, requirements are organized around objects, which integrate both data and functions. They are modeled after real-world objects that the system interacts with. In traditional analysis methodologies, the two aspects - functions and data - are considered separately.

Grady Booch has defined OOA as “Object-oriented analysis is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain”.

The primary tasks in the object-oriented analysis (OOA) are:

* Identifying objects
* Organizing the objects by creating an object model diagram
* Defining the internals of the objects, or object attributes
* Defining the behavior of the objects, i.e., object actions
* Describing how the objects interact

The common models used in OOA are use cases and object models.

**Object-Oriented Design**

Object-Oriented Design (OOD) involves the implementation of the conceptual model produced during object-oriented analysis. In OOD, concepts in the analysis model, which are technology−independent, are mapped onto implementing classes, constraints are identified and interfaces are designed, resulting in a model for the solution domain, i.e., a detailed description of how the system is to be built on concrete technologies.

**The implementation details generally include:**

* Restructuring the class data (if necessary),
* Implementation of methods, i.e., internal data structures and algorithms,
* Implementation of control, and
* Implementation of associations.

Grady Booch has defined object-oriented design as “a method of design encompassing the process of object-oriented decomposition and a notation for depicting logical and physical as well as static and dynamic models of the system under design”.

**Object-Oriented Programming**

Object-oriented programming (OOP) is a programming paradigm based upon objects (having both data and methods) that aims to incorporate the advantages of modularity and reusability. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs.

The important features of object-oriented programming are:

* The bottom-up approach in program design
* Programs organized around objects, grouped in classes
* Focus on data with methods to operate upon object’s data
* Interaction between objects through functions
* Reusability of design through the creation of new classes by adding features to existing classes

Some examples of object-oriented programming languages are C++, Java, Smalltalk, Delphi, C#, Perl, Python, Ruby, and PHP.

Grady Booch has defined object-oriented programming as “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 via inheritance relationships”.

**Object**

This is the basic unit of object-oriented programming. That is both data and functions that operate on data are bundled as a unit called as object.

**Class**

When you define a class, you define a blueprint for an object. This doesn't define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

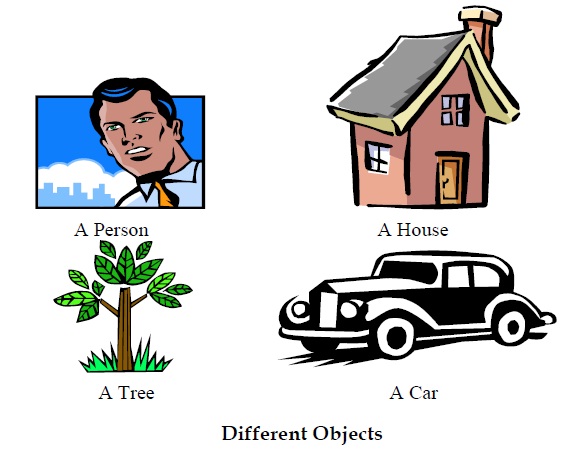
**Object-Orientation (OO)**

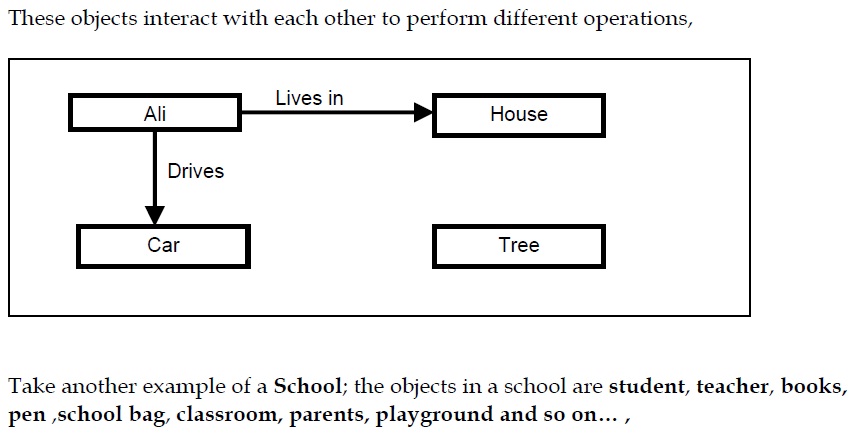
**What is Object-Orientation?**

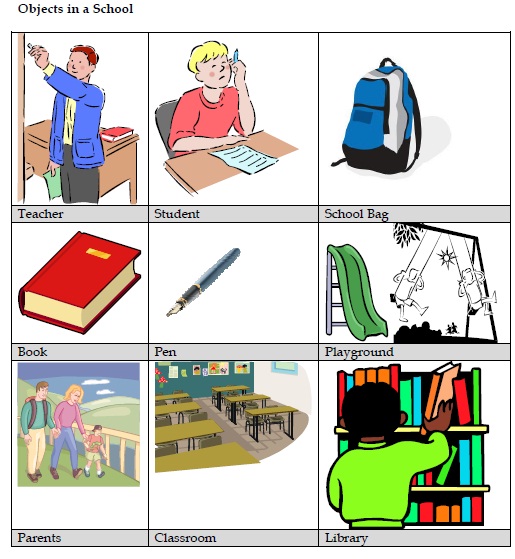
It is a technique in which we visualize our programming problems in the form of objects and their interactions as happens in real life.

**Examples:**

We have different objects around us in our real life that interact with each other to perform different operations for example:







Suppose we want to develop a fee collection system for a school for this we will need to find out related objects and their interactions as happens in real life.

In this way, we can say that **object orientation** makes it easier for us to solve real-world problems by thinking about the solution to the problem in terms of real-world objects. So we can say that in our daily life everything can be taken as an object that behaves in a certain way and has certain attributes. In object-orientation, we move our concentration to objects in contrast to the procedural paradigm in which we simply write our code in functions and call them in our main program.

**What is a Model?**

A model is an abstraction of something real or conceptual. We need models to understand an aspect of reality.

**Model Examples**

Highway maps

Architectural models

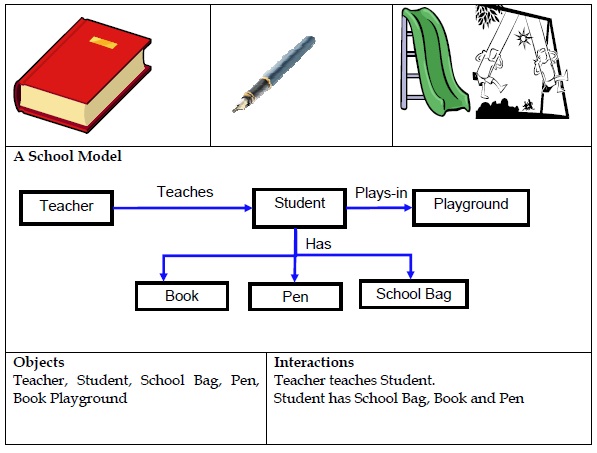
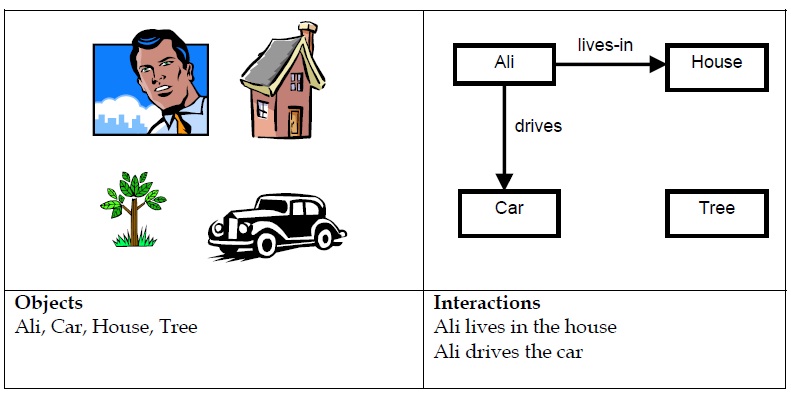
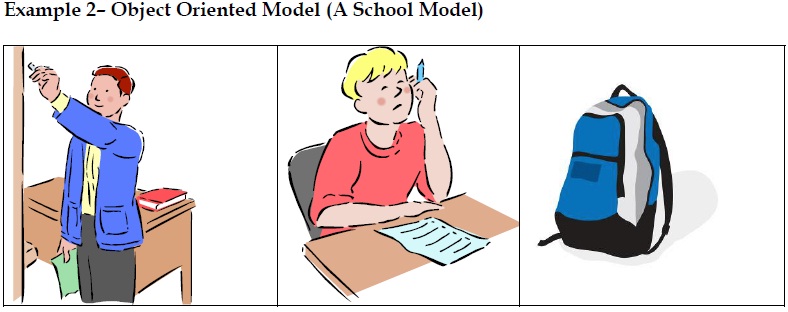
Mechanical models

**OO Models:**

In the context of programming, models are used to understand the problem before starting developing it. We make Object-Oriented models showing several interacting objects to understand

a system given to us for implementation.

**Example 1– Object Oriented Model**



**Object-Orientation - Advantages**

As Object-Oriented Models map directly to reality as we have seen in examples above therefore, We can easily **develop** an object-oriented model for a problem. Everyone can easily understand an **object-oriented model**.

We can easily implement an object-oriented model for a problem using an object-oriented language like c++ using its features1 like classes, inheritance, virtual functions and so on….

**What is an Object?**

An object is,

1. Something tangible (Ali, School, House, Car).

2. Something conceptual (that can be apprehended intellectually, for example, time, date and so on…).

An object has,

1. State (attributes)

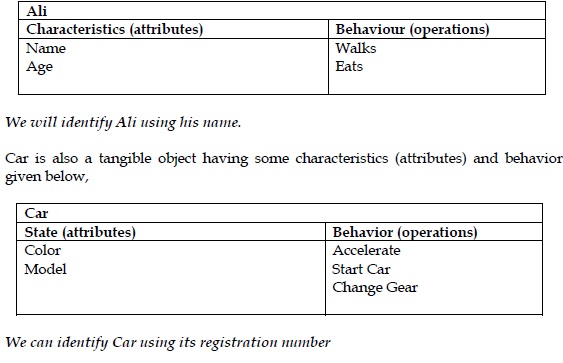
2. Well-defined behavior (operations)

3. Unique identity

**Tangible and Intangible Objects**

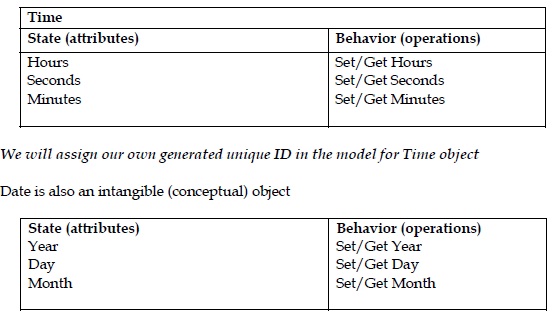
**Examples of Tangible Objects:**

Ali is a tangible object, having some characteristics (attributes) and behavior as given below,



**Examples of Intangible Objects (also called as conceptual objects):**

Time is an intangible (conceptual) object



We will assign our own generated unique ID in the model for Date object.