

# Introduction to Classes and Objects

## Introduction to Objects

- What is an Object?
  - Before diving into object-oriented Analysis, we should first discuss what actually an object is?
  - In this section I will start with the basic definition of an object, then I will refine this definition with the help of examples.
  - Generally, an object is anything that occupies space in the universe.
  - **For example:** The room you are sitting in contains several objects. The laptop you are working on is an object, the table you have kept your laptop on is an object. The chair on which you are sitting is also an object.



Laptop → Object



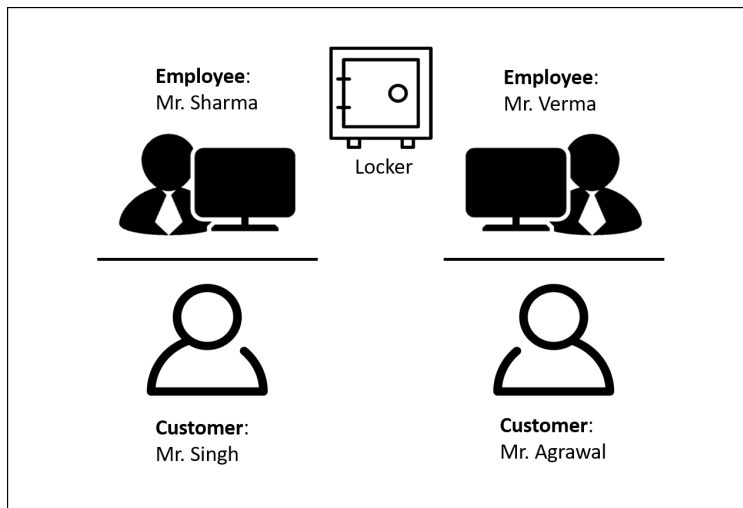
Table → Object



Chair → Object

- And, interestingly you are also an example of an object.
  - We can say, any tangible item which occupies space in the universe is an object.
  - Now a question must come to your mind what about intangible items? Can we have intangible objects?
  - I will try to explain this with the help of an example:
  - You just imagine that you are visiting your nearby bank. You are standing at the door of the bank. What things do you observe?
  - Definitely, you can see bank employees interacting with bank customers and the locker that the bank uses to keep the money. Also, you can notice that bank employees are using their computers to update the bank accounts of the customers.

- This is how a real bank looks like:



XYZ Bank

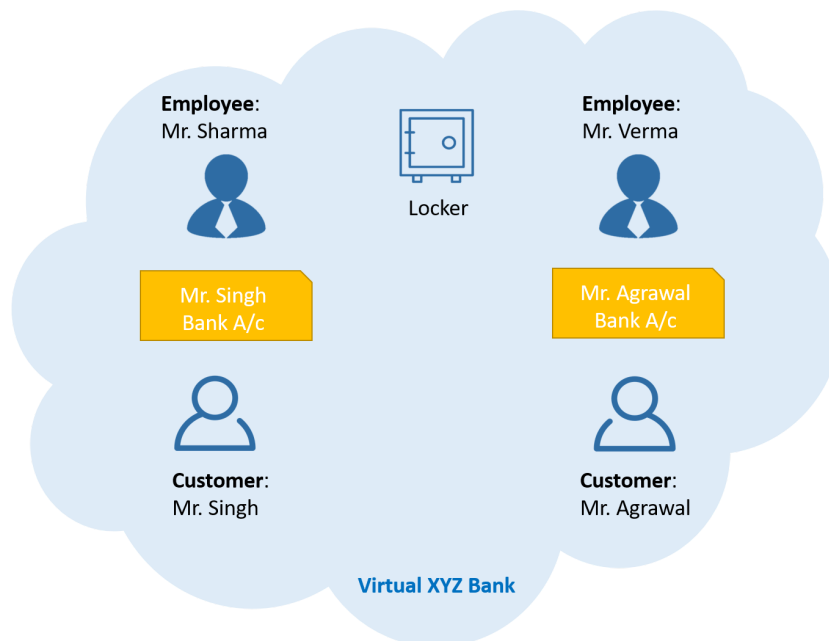
- Tangible Items:



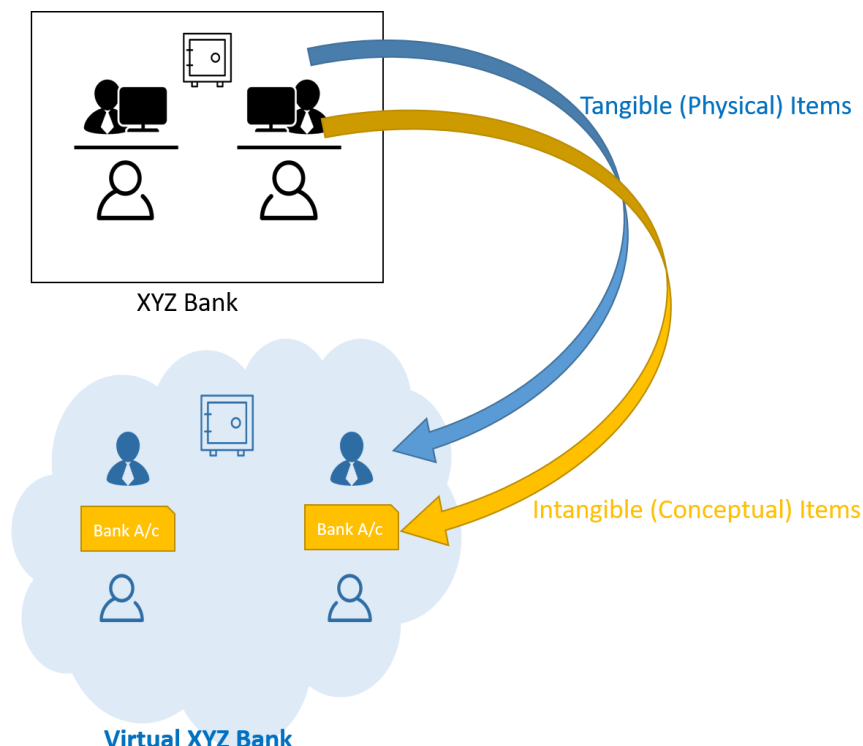
## Tangible Items

- Now just give a thought, how can I make this bank work virtually or in other words I want to implement a virtual bank on my computer which can do everything that a real bank can do.
- To achieve this, I first need to identify what all real-world entities need to be present in the virtual world.
- Let's try to move things one by one.
- Do we need the locker in our virtual bank? Yes, It is needed to keep track of the amount with the bank.
- Do we need employees? Yes because the administration and management of banks can be handled by employees only.
- Do we need customers? Yes because customers will interact with the bank accounts.

- Do we need personal Computers? Just Try to recall that Employees were using personal computers to update the bank accounts of the customers. So we don't need personal computers in the virtual world. But we will definitely require the bank accounts of customers in our virtual world.



- You can see that we have moved some physical and some conceptual items from the real-world to the virtual world to simulate the functionality of the real world.

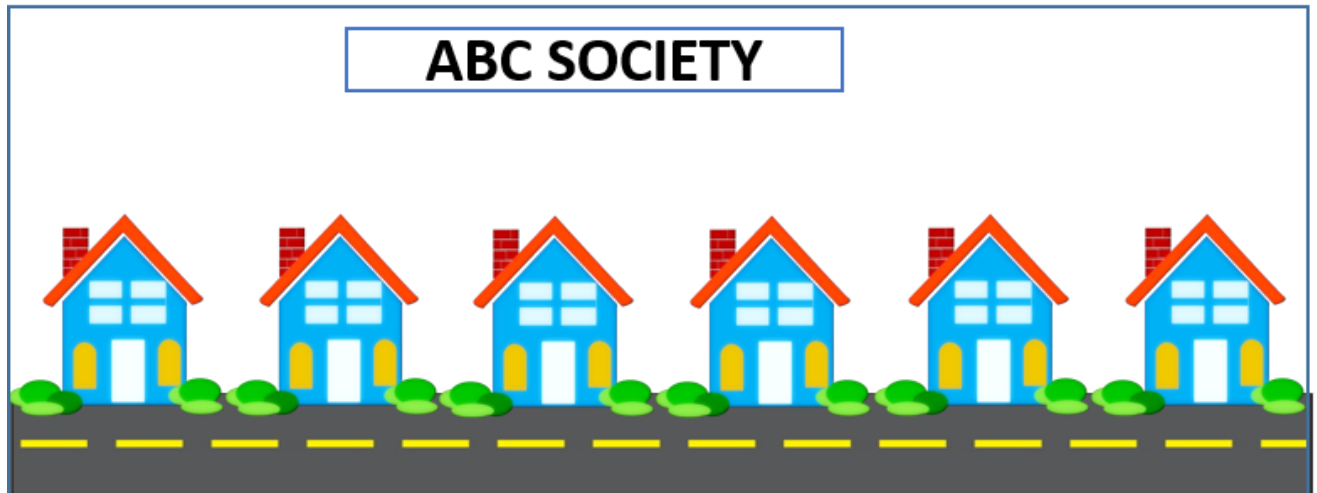


- Now we are at a point where we can give a more precise definition to the term object i.e.,

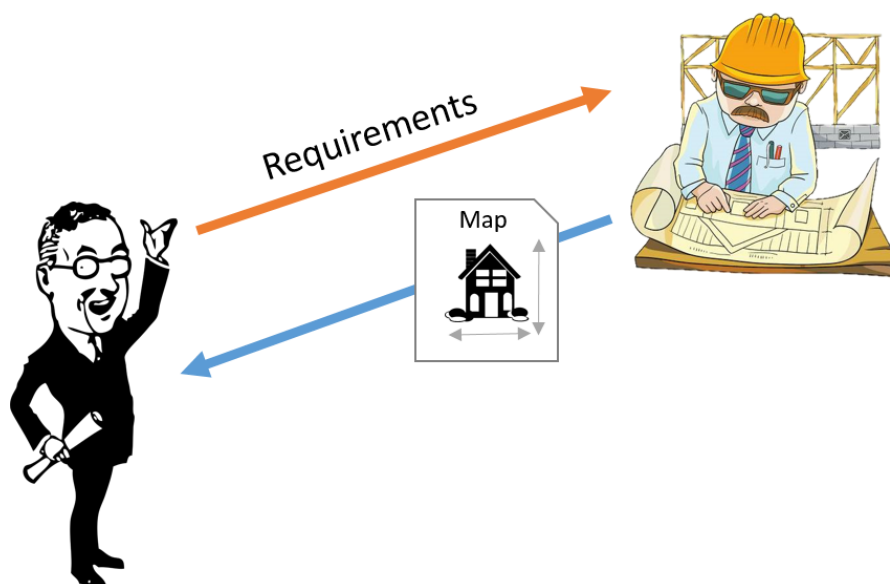
**An Object is a real-world element in an object-oriented environment that may have physical or conceptual existence.**

## Classes & Objects

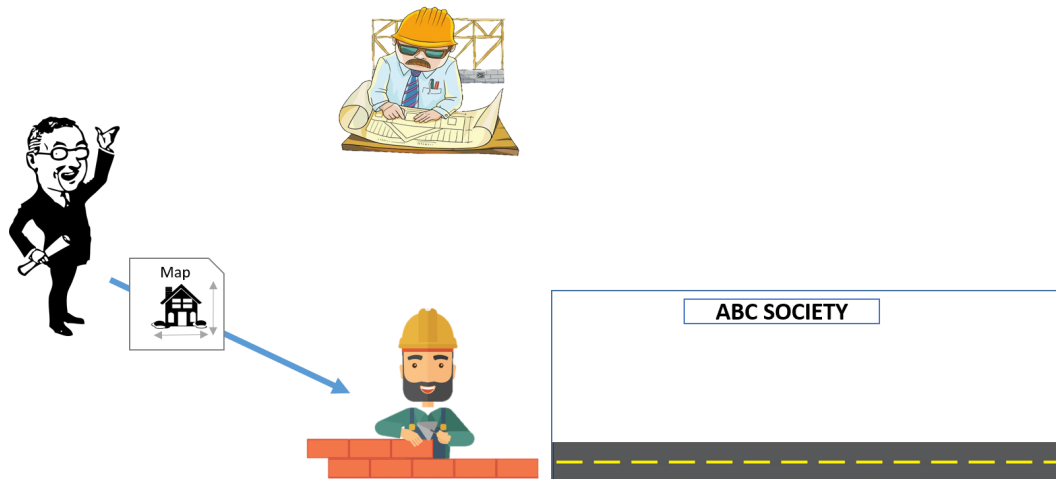
- In this section, we will help you to understand the concept of class with the help of some examples. Then we will go back to the bank example discussed in the previous section to filter out classes and objects in the virtual bank.
- If you have got a chance to visit any of the societies, you might have noticed that all the houses have the same structure. Have you ever wondered that how is it possible to create several similar houses?



- It is very simple:
  - The builder of society goes to the architect with his requirements.
  - On the basis of requirements, the architect creates a map containing all the details of the house.
  - Then, the builder hands-over this map to the contractor.
  - The contractor follows the details in the map while building each and every house; hence he is able to build several similar houses.
  - This map acts as a blueprint for the contractor.



- The same process is followed by a car company manufacturing cars. The company shares the design of the car with the manufacturing unit. The manufacturing unit follows the design or blueprints to build several cars with a similar structure.



- In both examples, we saw that blueprints contain all the information to build the house or to build the cars. But they themselves are neither the house nor the car.
- We can easily see, that the contractor used the blueprint to build n number of houses.
- Also, the manufacturing unit used the blueprint to build n number of cars.
- Now let's place all the blueprints on my right and all the instances on my left.
- We can say that all the instances occupy some space in-universe. Hence we can call them objects.
- Whereas, the blueprints on my right contains all the information to build these objects. These blueprints are also known as CLASSES.
- Finally, A class can be **defined as a blueprint or description of objects that can be created from it**. We have also observed the following properties of a class:

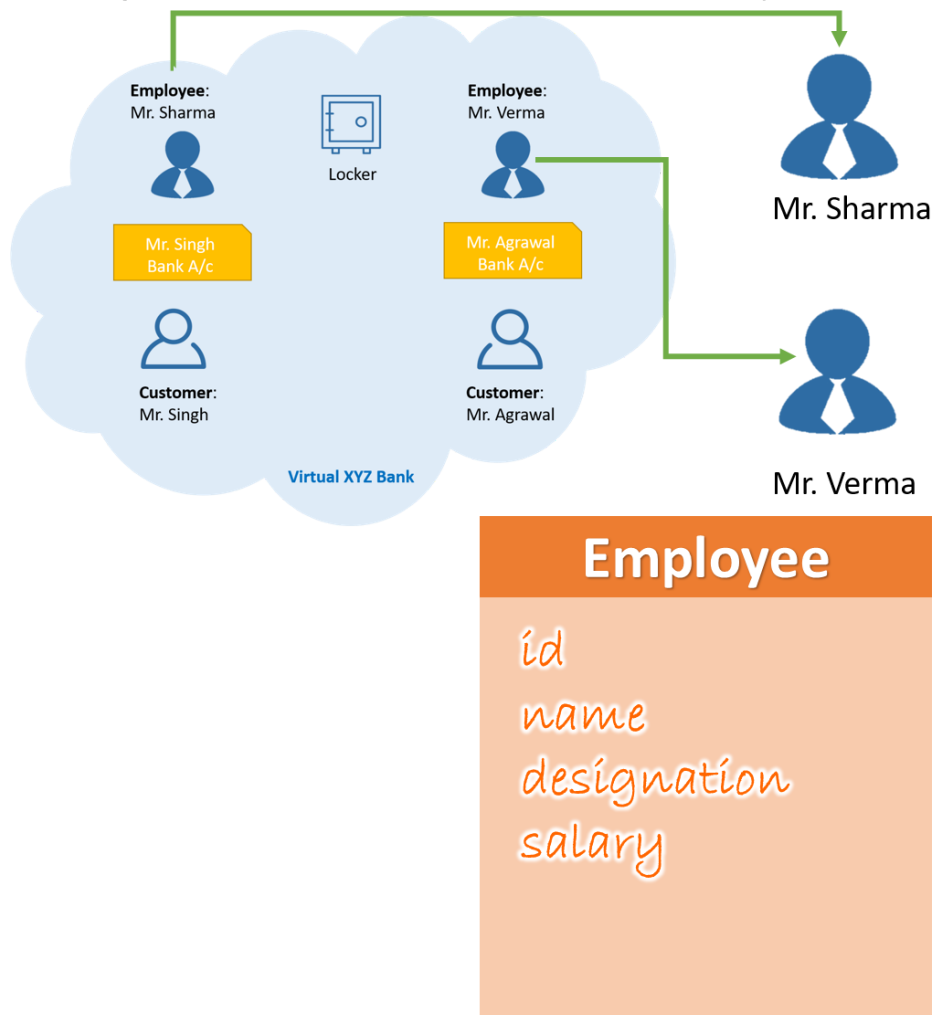
<b>Blueprints</b> Or <b>Classes</b>	<b>Instances</b> or <b>Objects</b>
---	--



1. NOT REAL: Class is not a REAL THING; it's just a concept, actually the objects created from a class are real.
2. A single class can be used to create any number of instances or objects.

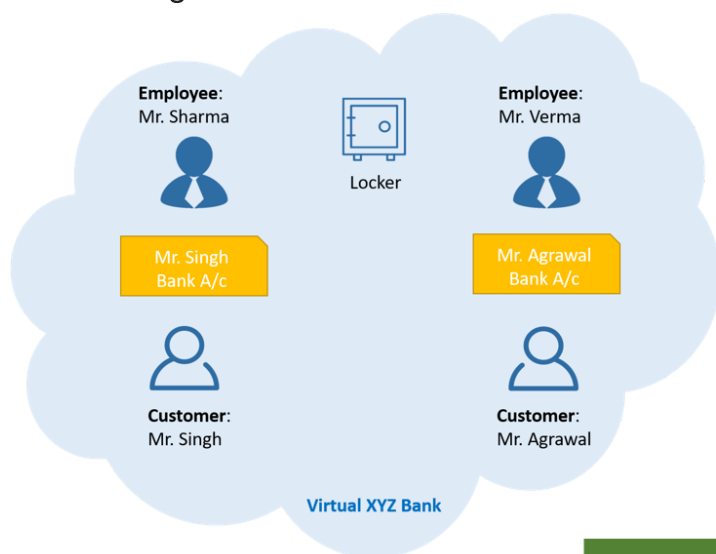
## Virtual Bank - Classes & Objects

- It's time to look back at the example of virtual banks and refine our concept of classes and objects.
- Here it is,
- Now we will perform a small activity to classify similar objects in our virtual bank.
- Let's pick Mr. Sharma and Mr. Verma first.
- Now because they both are employees, we will write down the things that they have in common
- For example, they will have an employee id.
- One thing to note here is that Mr. Sharma will have a different employee-id from Mr. Verma, But they both will have some employee id.
- Similarly, they will have some designation and of course salaries.
- Now we have a set of attributes (properties) that an employee will always have, so we will put a tag over these attributes and name it as Employee.



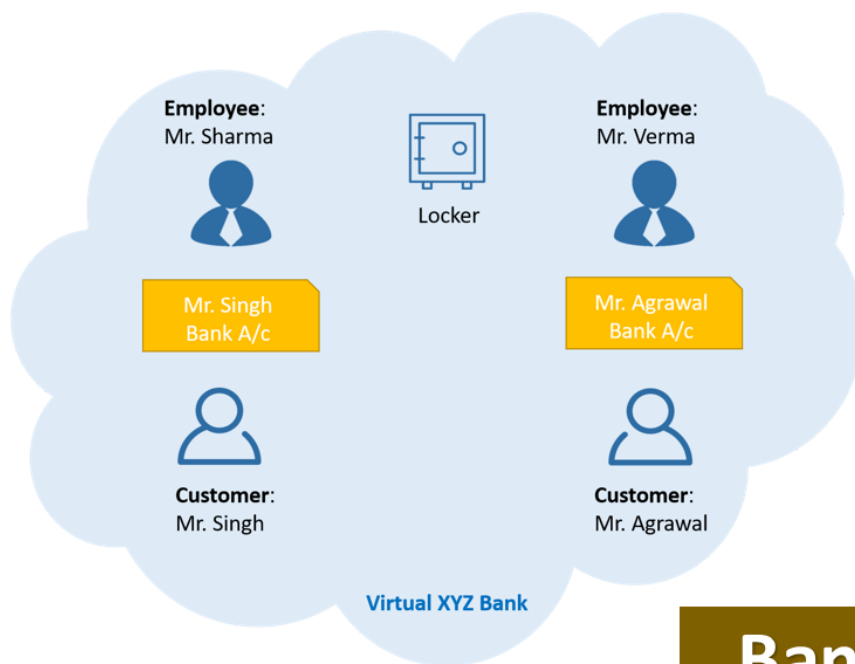
- Can you guess, what actually we have done?
- Yes, we have created a blueprint or description for employees. Or we can say, we have created an employee class, and Mr. Sharma and Mr. Verma are objects of Employee class.
- Also, we can create as many objects as we want from this employee class.
- All newly created employees will have the same properties as possessed by Mr. Sharma and Mr. Verma.
- In the same way, let's now pick Mr. Agrawal and Mr. Singh.

- Their common attributes can be customer\_id, account\_number, date\_of\_birth, and address.
- Let's put a tag over these attributes as CUSTOMER.
- Now every new customer will also have the same properties as possessed by Mr. Agrawal and Mr. Singh.



- So far we have created the blueprints for EMPLOYER and CUSTOMER. Let's create a blueprint for BANK\_ACCOUNT as well.

- The BANK\_ACCOUNT class will look like this and will have account\_number, account\_holders\_name, the BALANCE in the account, TYPE of account and also it can have the STATUS of the account.



Mr. Singh  
Bank A/c

Mr. Agrawal  
Bank A/c

Bank_Account
number
holder_name
type
balance
status