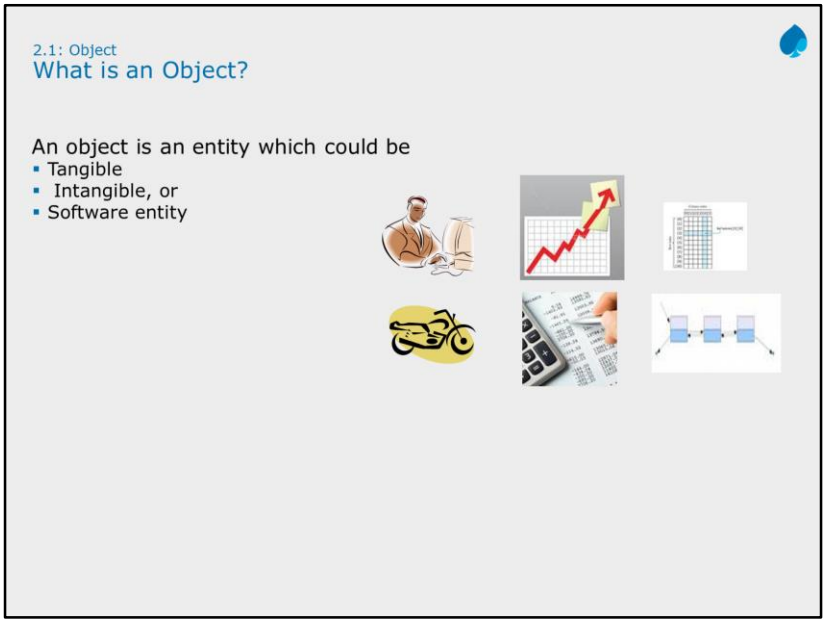

Core Java 8
Lesson 2: Objects and Classes
Capgemini

Lesson Objectives

In this lesson, you will learn:

- What is an Object?
  - State, Behavior, and Identity of an Object
- What is a Class?
  - Attributes and Operations of a Class
  - Access modifiers and its types
  - Constructors and Destructors
  - Static members

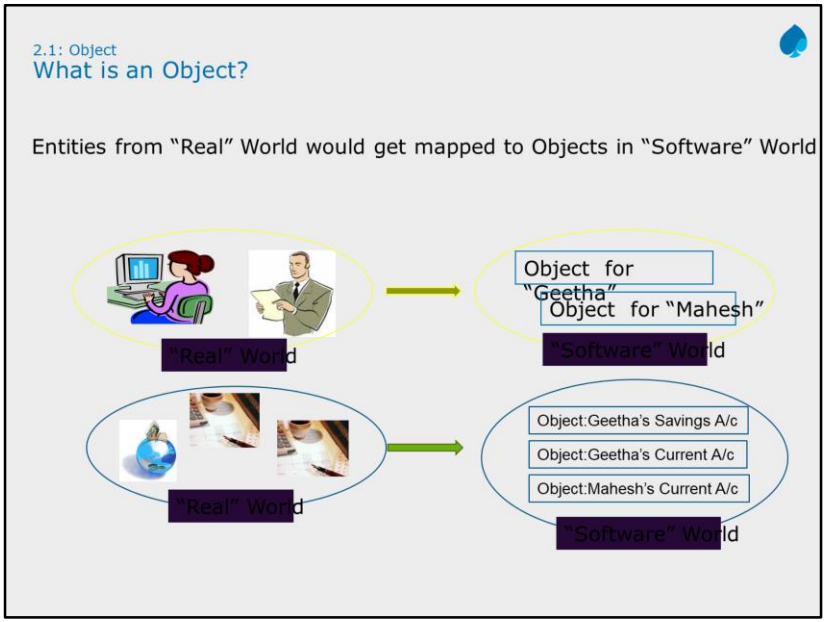


What is an Object?
An object in the real-world, can be physical, conceptual, or a software entity. We come across so many objects in the real world. In fact, everything can be considered as an object - a person, a pen, a vehicle, a book, etc. Essentially these are all tangible things that exist, can be felt, or can be destroyed. However, there could be other entities which may not be considered as objects in the real world, like an account or a contract, or a set of business charts, or a linked list since they are “intangible” or “conceptual”. Nevertheless, they also have a well defined structure and behavior, and hence are treated as objects in the software domain.

Examples of tangible entities
Person, Pen, Vehicle

Examples of intangible entities
Account, Contract, Business Charts

Examples of software entities
Database Management System



Remember the scenario from Banking System? "Geetha" and "Mahesh" from our example of Banking System are entities in the Real World. These would get mapped into corresponding objects in the software world. So in our OO application, we will have an object corresponding to "Geetha" and another object corresponding to "Mahesh". Similarly, we would have objects corresponding to Geetha's Savings Account, Geetha's Current Account as well as Mahesh's Current Account.

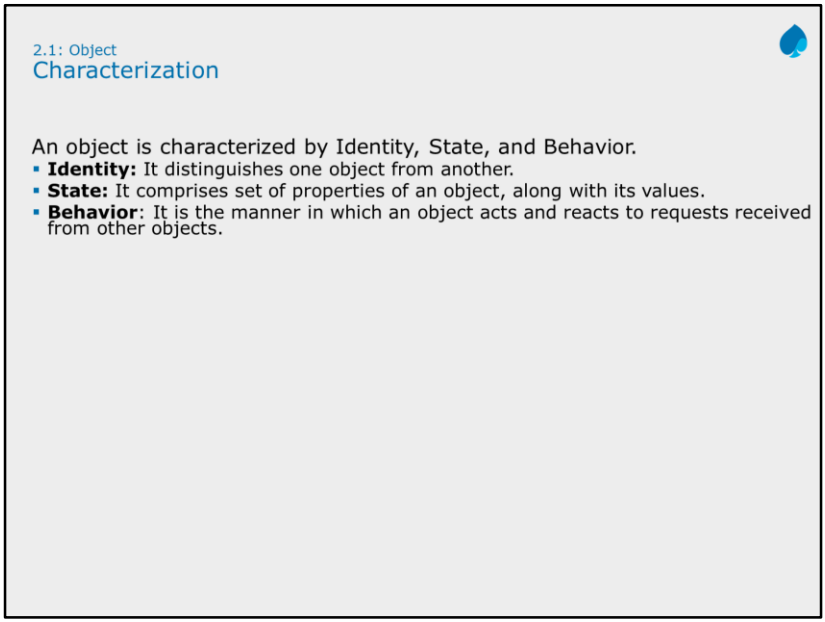
Typically objects could correspond to following

Roles played by people interacting with system (Like Geetha and Mahesh who are "Customer" objects)

Structures used for storing and processing data (Like Account objects for Geetha and Mahesh)

Other systems or devices interacting with system (Like Utility Payment System that can be accessed from Bank System)

Events and entities of the system (Like a transaction or a account status report)



Each object is characterized by identity, state, and behavior.

Identity: Two books of same title are still two different books - they are two instances of a "book" which happen to have similar properties, just as there will be two copies if they existed in the library. The identity of one is to be distinguished from the other.

State: It is one of the possible conditions that an object may be in. It is indicated by the set of values that each of its attributes possesses.

For example: An account object may be in an active or suspended state depending on the balance that it possesses.

Behavior: It is what an object does when it receives instructions. For example: Deposit or withdrawal that occurs against an account object.

2.1: Object

Object State

State of an object is one of the possible conditions in which the object may exist.

Bank Account Modeled as a Software Object

Attributes of the object:
AccountNumber: A10056
Type: Savings
Balance: 40000
Customer Name: Sarita Kale

The state of an object is not defined by a "state" attribute or a set of attributes. Instead the "state" of an object gets defined as a total of all the attributes and links of that object.

What is an Object? – Object State:
The current state of an object is defined by the set of values of its attributes and the links that the object has with other objects. The current state of an object is said to have changed, if one or more attribute values change. The object remains in control of how the outside world is allowed to use it:

- by assigning a state (e.g., account number, Account type, balance) to itself, and
- by providing methods for changing that state

2.1: Object
Object State

Behavior of the Object depends on the State of the Object

Withdrawal depends on the State of the Account Object

State where withdrawal is permitted

Attributes of the object:
AccountNumber: A10056
Type: Savings
Balance: **40000**
Customer Name: Sarita Kale

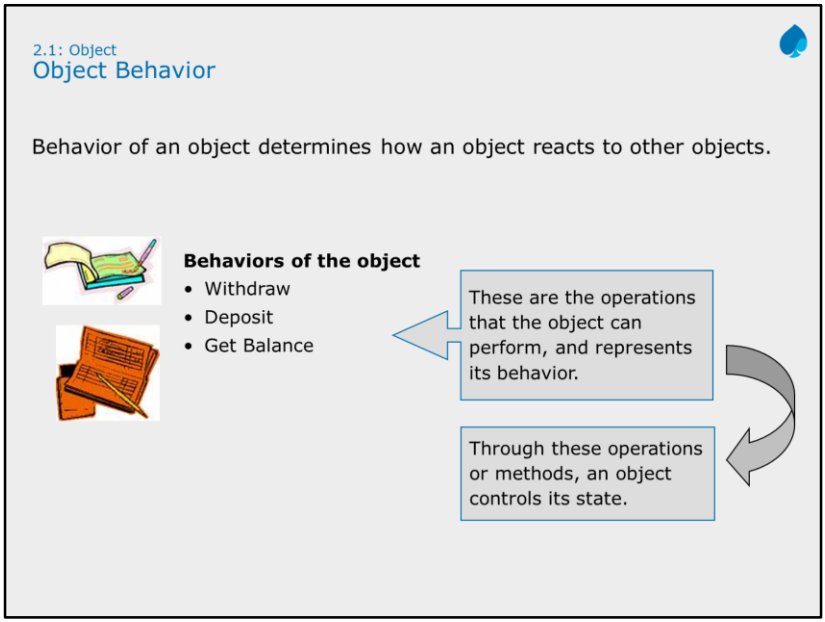
State where withdrawal is NOT permitted

Attributes of the object:
AccountNumber: A10056
Type: Savings
Balance: **500**
Customer Name: Sarita Kale

How many States of an Object should we consider??

The behaviour of an object in terms of how an object responds, depends on state of object. If the bank has a business rule on the minimum account balance, then an operation such as withdrawal will not be permitted if the balance is less than what is permitted.

At any point, an object will be in a single state. As such, any new combination of attribute values would imply a new state for an object. That means there could be infinitely many states for each object. Should we consider each of these states?? Well No! We only need to consider the object states which will have an impact on the behaviour of the object.



What is an Object? – Object Behavior:
Behavior is what an object does on receiving a set of instructions. Object behavior is represented by the operations that the object can perform.
For example: A Bank ATM object will have operations such as withdraw, print transactions, swipe card, and so on. A bicycle object may have operations such as change gear, change speed, and so on.

2.1: Object

Object Identity

Two objects can posses identical attributes (state) and yet have distinct identities.

What is an Object? – Object Identity:
Two accounts may possess same attributes like type, balance, etc. Yet these two accounts have separate distinct identities. One account could be mine, and the other could be yours. Although objects may share the same state (attributes and relationships), they are separate and independent objects with their own “unique identity”.

2.2: Class
What is a Class?

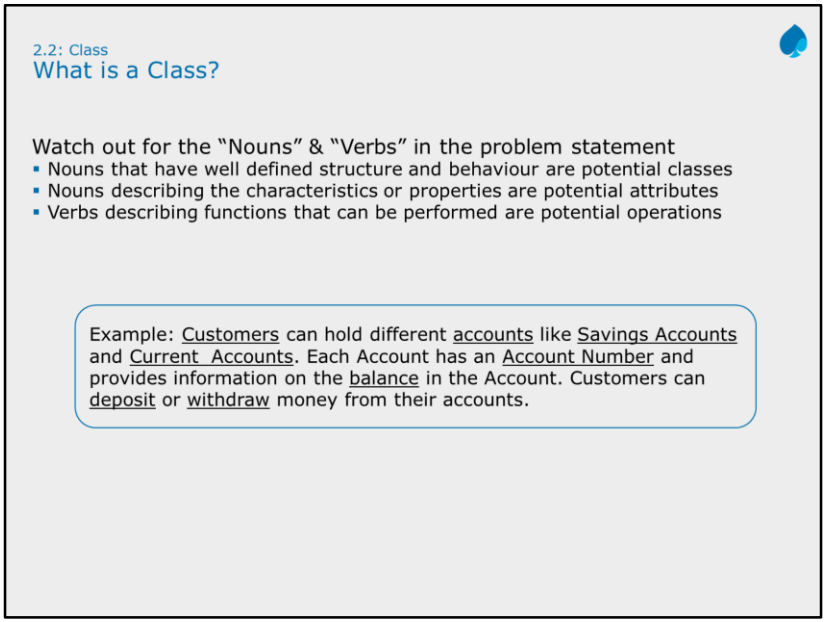
A Class characterizes common structure and behavior of a set of objects.
It constitutes of Attributes and Operations.
It serves as a template from which objects are created in an application.

Class name	Customer
Class attributes	Name, Address, Email-ID, TelNumber
Class operations	displayCustomerDetails() changeContactDetails()

What is a Class?
Classes describe objects that share characteristics, methods, relationships, and semantics. Each class has a name, attributes (its values determine state of an object), and operations (which provides the behavior for the object).
What is the relationship between objects and classes?

What exists in real world is objects. When we classify these objects on the basis of commonality of structure and behavior, what we get are classes. Classes are “logical”, they don’t really exist in real world. While writing software programs, it is the classes that get defined first. These classes serve as a blueprint from which objects are created.
For example: In the example shown in the slide, there may be thousands of bank customers all having same set of attributes (i.e., Name, Address, Email-ID, TelNumber). Each customer is created from the same set of blueprints, and therefore contains the same attributes. Similarly, there can be thousands of Bank Accounts instantiated from the same “Account” class!
In terms of Object-Oriented technology, we say that these customers are all “instances” of the “class of objects” known as Customer. A “class” is the

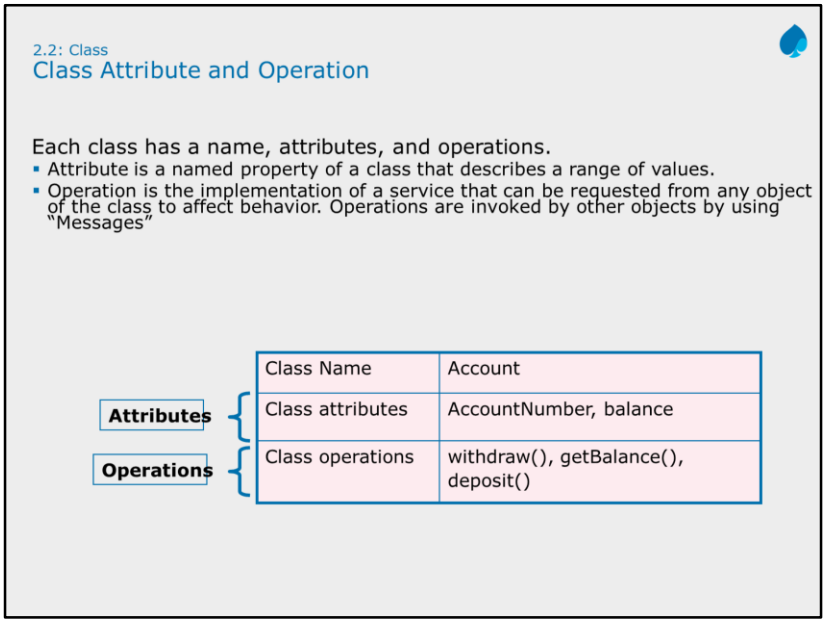
blueprint from which individual “objects” are created.



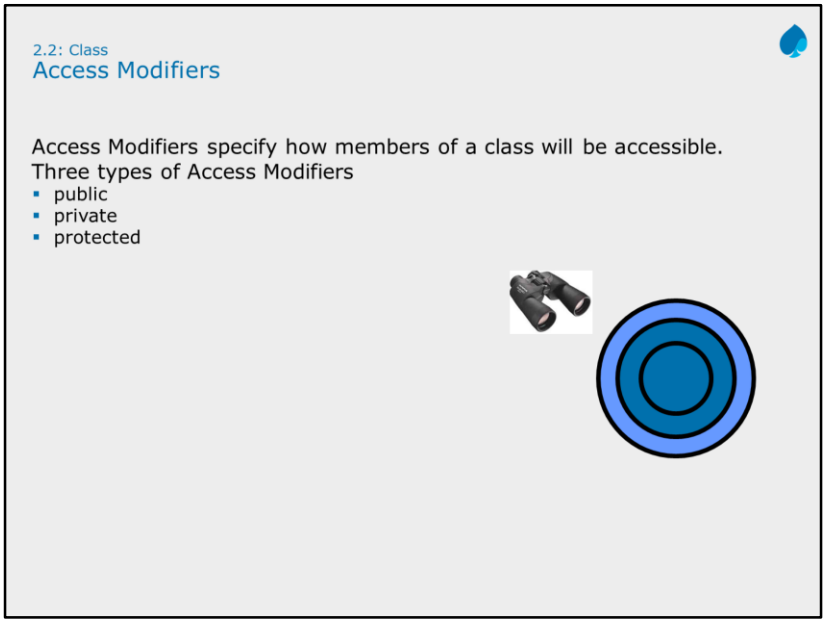
To help identify potential classes, their attributes and their operations, watch out for the nouns and verbs of the problem statement. A Noun having a well defined structure and behaviour, which can be a standalone entity, is a potential class. Nouns which cannot be a stand alone entity but point to properties or characteristics of something could be potential attributes. And finally, verbs describing what could be "done" are potential operations of the class.
In the example here, note that all nouns and verbs are underlined. Potential Classes could be Customer, Account, Savings Account, Current Account. Potential Attributes (of Account Class) could be Account Number and Account Balance. Potential operations (of Account Class) could be deposit and withdraw.

2.2: Class
Lab

Objects and Classes
- Labs 1.1 and 1.2



Getting into Details – Class Attribute and Operation:
A class has named properties, which are attributes of the class. An attribute would be of a specific type. At runtime, an object will have associated values for each of its attributes.
A class can have several operations. An operation is an implementation of a service that can be requested from an object.
When an operation of an object has to be invoked by another object, it passes a "message" to the object. Messages would correspond to the operation name.



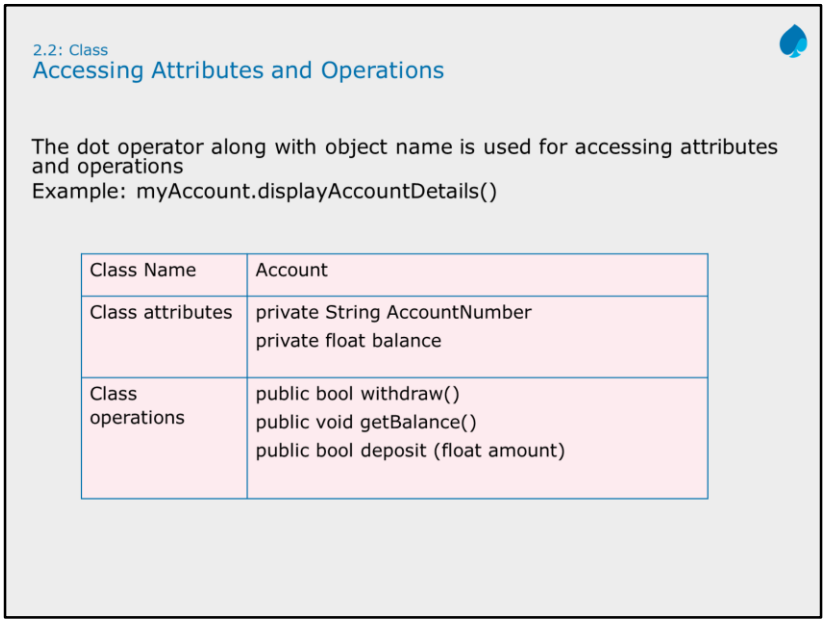
Getting into Details – Access Modifiers:
Access Modifiers specify how members of a class will be accessible. OOP supports the following three types of access modifiers:

Public means accessible to all. An object can access the public variable outside its own class.

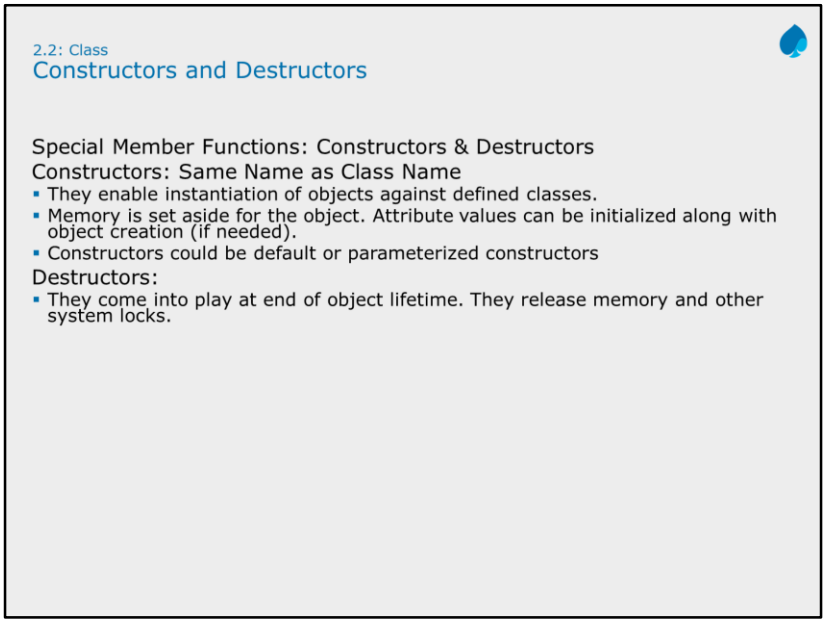
Private means accessible only to the own class. An object can access the private variable only in its own class.

Protected means accessible to own class and its subclass. An object can access the protected variable only in its own class or its subclass. Concept of subclass is discussed later.

Information Hiding means hiding all the details of an object that do not contribute to its essential characteristics. To this end, access modifiers help to restrict the accessibility to attributes and operations.

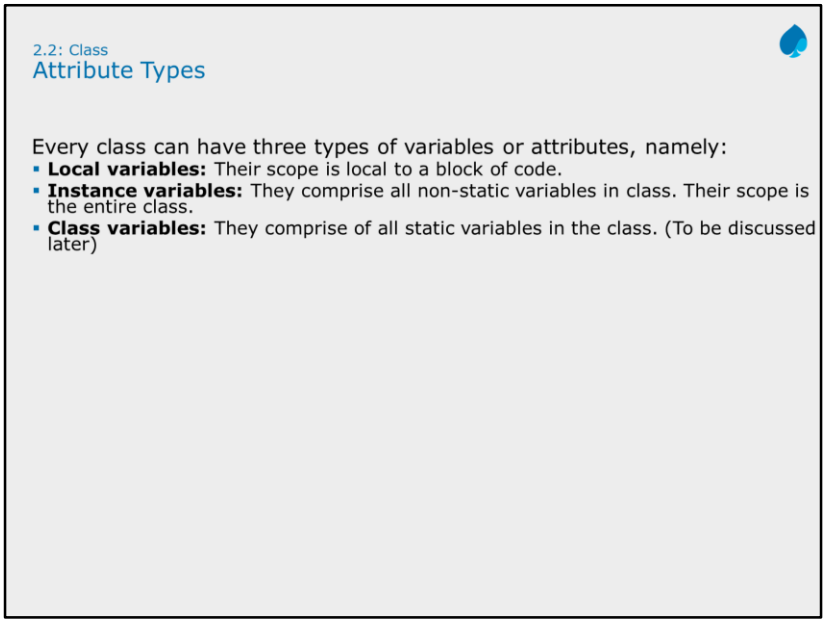


Getting into Details – Access an attribute or an operation:
To access members, a dot operator is used against the corresponding Account object.
For example: If MyAccount is an object of Account type, to display the details of this object, we would specify MyAccount.displayAccountDetails().
From where an attribute or operation can be accessed depends on the access modifiers. For eg. if the Account Class had a private operation for calculating interest, this operation can only be invoked from within another operation of the same class.
Both attributes of the Account Class are private here, which means they are not accessible from outside this class.



Getting into Details – Constructors and Destructors:
Objects are created using Constructors. The Constructors are special member functions of the class that share the same name as that of the class. When objects are created, memory is set aside for them. The attribute values of objects may be initialized, if needed. A class may have multiple constructors since we may want objects to be created and attributes initiated in various ways. The constructor that takes no parameters is a default constructor, while constructors that take 1 or more parameters are the parameterized constructors.

When objects are no longer needed, Destructors come into play. The most common use of destructors is to de-allocate memory that was allocated for the object by the Constructor. There can be only 1 destructor for a class. The destructor name is same as class name, and preceded with tilde (~) sign. Eg. ~Account().



Getting into Details – Attribute Types:
Local variables: These are variables declared within a function, or method, or any block of code, and the arguments passed to a function. They are not accessible outside the block in which they are defined.
Instance variables: These are declared as a part of class. We have seen examples of such variables earlier – AccountNumber, Balance, etc. These are accessible by all the functions written within the class.
Class variables: They are also called Static variables. (This will be covered in detail later) Such variables are useful when all the objects of the same class need to share a common item of information.

2.2: Class
Lab

Objects and Classes
- Labs 2.1 and 2.2

Summary

In this Lesson, you have learnt that:

- Objects have an identity, state, and behavior.
- Class is a user-defined description of set of objects, sharing same structure and behavior.
- Access modifiers help to restrict the accessibility.
- Constructors and Destructors help in memory allocation and de-allocation, respectively

Answers to Review Questions

Question 1: Behaviour

Question 2: True

Question 3: Static

Review Question

Question 1: ___ determines how an object reacts to other objects.

- Option 1: State
- Option 2: Behavior
- Option 3: Identity
- Option 4: Attribute

Question 2: A class can have zero or more number of operations.

- True / False

Question 3: ___ variables are accessible by all the functions written within the class.

Review Question: Crossword

	1	2	3	4	5	6	7	8	9	10	11
1		O						C			
2											
3											
4											
5											
6											
7											
8					e						
9											
10											
11											
12											
13											
14											

Clues. 14 rows (1-14) , 11 Cols (1-11), (Row,Col) Combination
Across:
1-1: Objects attributes can be initialized using this special function (11)
8-1: A contract is a _____ type of an entity (10)
12-1: This data is allocated once and shared among all object instances of the same type(6)
14-2: Named property of a class that describes a range of values. (9)

Below:
1-2: Implementation of a service (9)
1-4: This describes the possible conditions that an object may be in (5)
1-8: Blue print of an object (5)
7-5: Operations the object can perform (8)