

Core Java 8

Lesson 04: Some More
Concepts in OOP



Lesson Objectives

In this lesson, you will learn about:

- Static Members
- Abstract Classes
- Interfaces
 - Interface versus Abstract Classes
- Packages





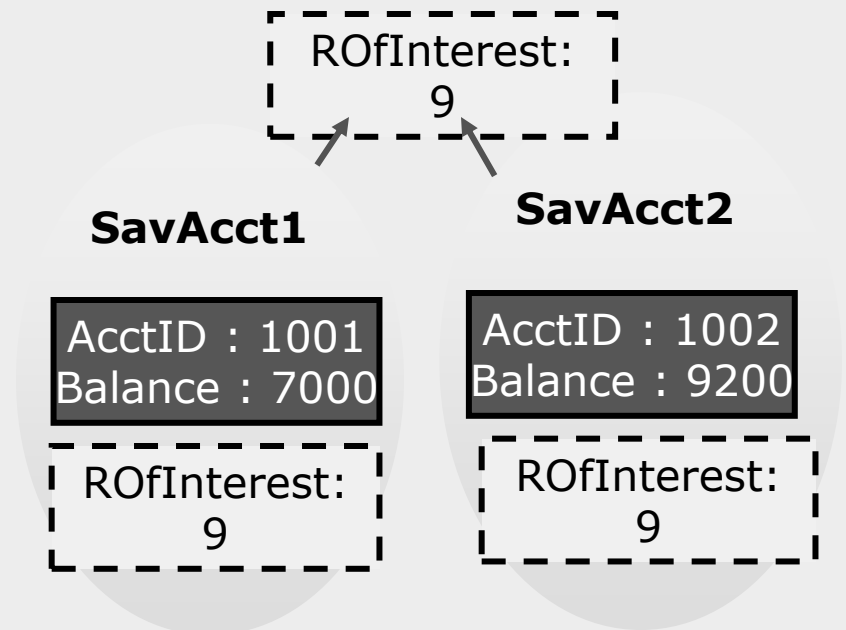
4.1: Static Members

Introduction

Each object has its own copy of data (instance variable)

Use Static (Class variable) if data members are to be shared amongst all object instances of the given type

- Example: Rate of Interest for Savings Account will be the same in the bank for all Accounts, so common copy for data is sufficient. Not so for Account Balances!





4.1: Static Members

Introduction

Static Member Functions can be invoked without an object instance.

- For example: Counting the number of Customer Objects created in the Banking System – this is not specific to one object!

Example

```
class Customer{  
    static int customerCount;  
    Customer(){  
        customerCount++;  
    }  
    static int countCustomers(){  
        return customerCount;  
    }  
}
```

Invoking above static function would be like

```
Customer.countCustomers();
```



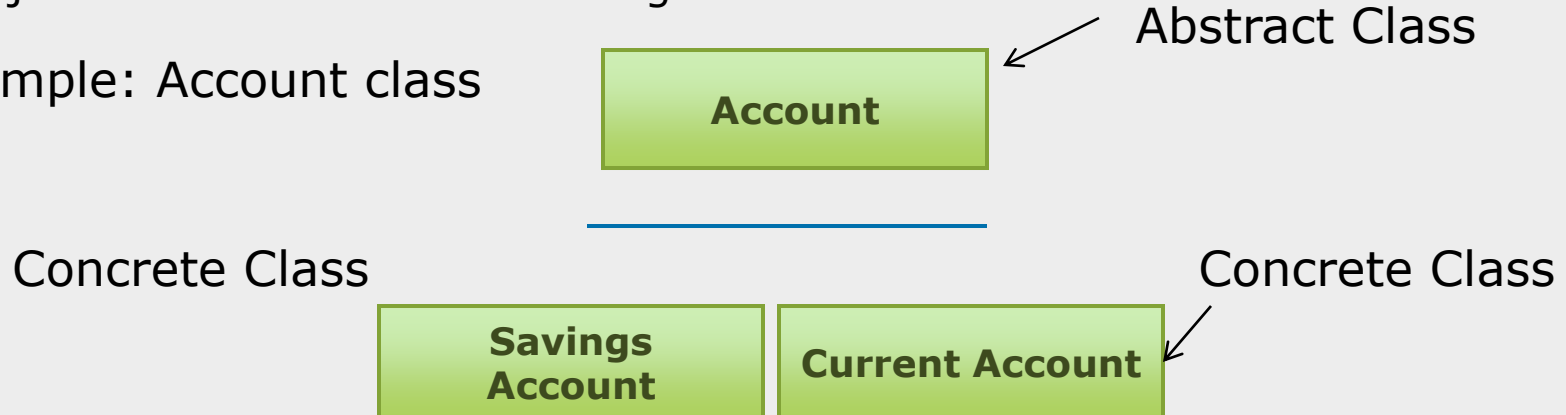
4.2: Abstract Class

Concept of Abstract Class

Abstract class is a Special Type of Base Class.

- Implementation details are undefined for one or more operations, they are implemented by derived classes.
- Objects cannot be instantiated against such classes.

Example: Account class





4.2: Abstract Class

Concept of Abstract Class

Why Abstract Classes?

- Establish Structure to Class Hierarchies:
 - They capture the commonality amongst hierarchy of classes, at the same time “changing” implementations can be left to derived classes.

Design Heuristic: Base of a class hierarchy should be an abstract class



4.2: Abstract Class

Lab

Abstract Classes and Polymorphism

- Lab 4.1





4.3: Interface

Concept of Interface

Interface is a specification of a set of operations that indicates service provided by a class or component.

- Interfaces have only specifications, no implementations.
- Implementations are provided by classes that implement the interfaces.

Why Interfaces?

- They allow polymorphism – without being restrictive about class hierarchies.
- They enable Plug and Play architecture



4.3: Interface

Abstract Class versus Interface

Abstract Class	Interface
No instantiation	No instantiation
Can have implementations for some operations	Purely specifications, No implementations
Can encapsulate a common behaviour for use by related classes	Can encapsulate a common behaviour for use by unrelated classes
Class derived from Abstract classes can provide implementations for only some operations if needed	Class or component implementing an Interface is bound to provide implementation for ALL operations specified in interface
Widely supported by OO Programming Languages	No support for this concept in some of the earlier OO Programming Languages



4.4: Packages

Concept of Packages

A package is a logical grouping unit of related classes and interfaces.

- Unique identifiers for classes and interfaces are needed in same package.
- A package maps to directory structure for application development.

Why Packages?

- It facilitates maintenance of large systems by partitioning the set of classes into **manageable chunks**.



4.4: Packages

Lab

Consolidated Exercise

- Lab 5.1





Summary

In this lesson, you have learnt that:

- Static Members allow sharing of data across different objects of the same class
- Abstract classes establish structure and meaning to code.
- Interfaces formalize polymorphism.
- Packages are a logical grouping mechanism





Review Question

Question 1: Choose right options. Abstract classes

- Option 1: Members have implementations
- Option 2: Sit toward the top of a class hierarchy
- Option 3: Establish structure and meaning to code
- Option 4: Have information and behavior common to all derived classes
- Option 5: Can be instantiated



Question 2: Abstract classes allow you to partially implement your class.

- True / False

Question 3: ____ provides mechanism to logically group classes.

Review Question: Crossword



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
3											p				
4															
5															
6					R										
7															
8															
9															
10															
11															

