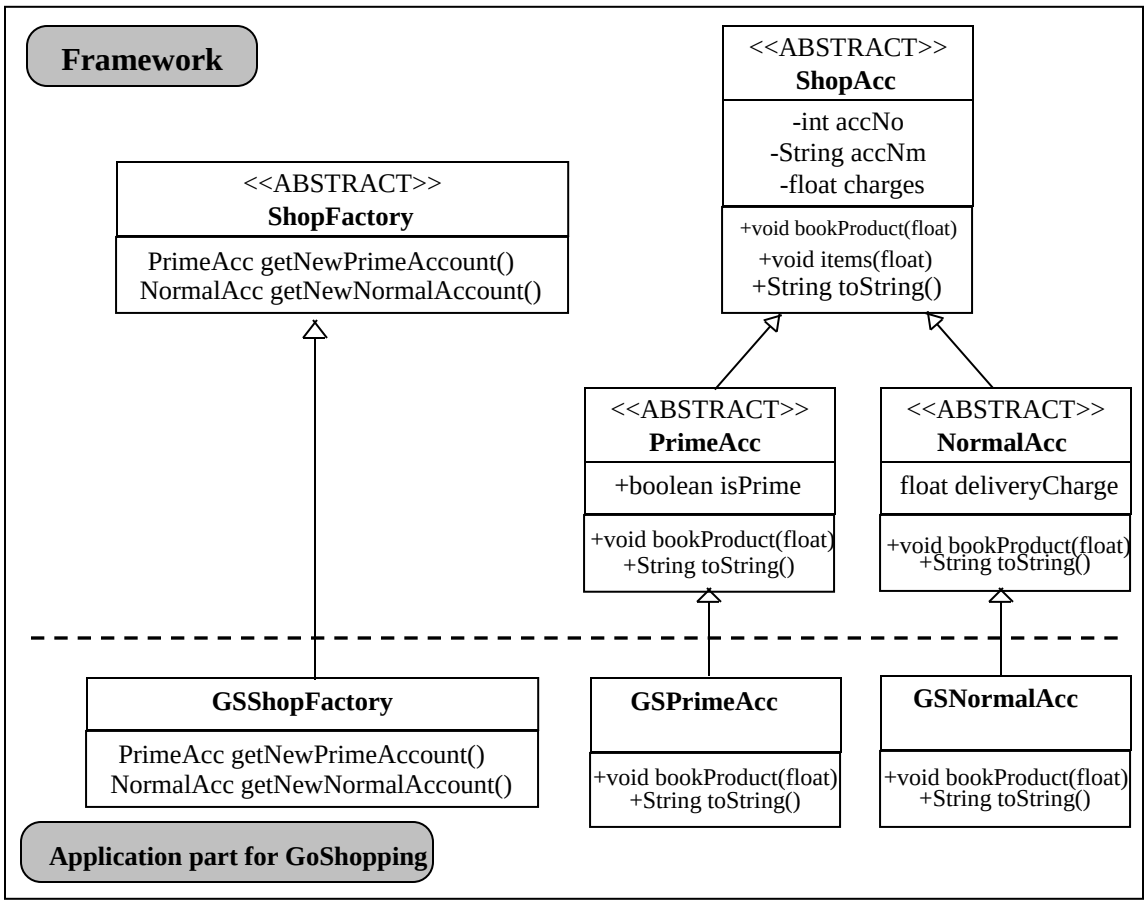
Java Case Study – I
Framework for Online Shopping Application

Design a simple framework for Online Shopping Application to represent Prime Accounts and Normal(non-prime) Accounts.

Use the framework to design application for GoShopping App.

Objectives

- To understand the concept of framework in application development.
- Areas of application for Abstract classes, abstract methods etc.
- Polymorphism and its uses,
- Final fields and Lazy Initialization
- Getter and Setter methods
- Lazy Binding of methods



Design following class structures for framework…

1. Abstract ShopAcc: An abstract class to represent a Online Shopping account.

Fields	Access	Type	Property
accNo	private	int	Read Only
accNm	private	String	Read-Write
charges	private	float	Read Only

Constructors	Access	Parameters
	Public	AccNo, accNm, charges

Methods	Access	Return Type	Parameters	Particulars
bookProduct	public	void	float	
items	public	void	float	
toString	public	String		Overridden

2. Abstract PrimeAcc extends ShopAcc: An abstract class to represent specific case of Prime Account. It extends ShopAcc. The overridden bookProduct method of the class must not allow extra charges for delivery in the account.

Fields	Access	Type	Property	Particulars
isPrime	private	boolean	Read Only	
deliveryCharges	private	float	Read Only	static final

Constructors	Access	Parameters
	Public	AccNo, accNm, charges, isPrime

Methods	Access	Return Type	Parameters	Particulars
bookProduct	public	void	float	Overridden
toString	public	String		Overridden

3. Abstract NormalAcc extends ShopAcc: An abstract class to represent specific case of Normal Account. It extends ShopAcc. The overridden bookProduct method of the class must allow delivery charges for the account.

Fields	Access	Type	Property	Particulars
deliveryCharges	private	float	Read Only	final

Constructors	Access	Parameters	Particulars
	Public	AccNo, accNm, charges, deliveryCharges	Lazy initialization for creditLimit.

Methods	Access	Return Type	Parameters	Particulars
bookProduct	public	void	float	Overridden
toString	public	String		Overridden

4. Abstract ShopFactory: An abstract class having necessary factory methods to instantiate new Prime or Normal types of accounts.

Methods	Access	Return Type	Parameters
getNewPrimeAcc	Public	PrimeAcc	AccNo, accNm, charges, isPrime
getNewNormalAcc	Public	NormalAcc	AccNo, accNm, charges, deliveryCharges

Design following class structures for application part…

5. Concrete GSPrimeAcc: A concrete class representing online shopping specific Prime Account. It extends PrimeAcc.

Fields	Access	Type	Property	Particulars
Charges	private	float	Read Only	static final

Constructors	Access	Parameters
	Public	AccNo, accNm, charges, isPrime

Methods	Access	Return Type	Parameters	Particulars
bookProduct	public	void	float	Overridden
toString	public	String		Overridden

6. Concrete GSNormalAcc: A concrete class representing online shopping specific Normal Account. It extends NormalAcc.

Constructors	Access	Parameters	Particulars
	Public	AccNo, accNm, charges, deliveryCharges	Lazy initialization for deliveryCharges.

Methods	Access	Return Type	Parameters	Particulars
bookProduct	public	void	float	Overridden
toString	public	String		Overridden

7. Concrete GSShopFactory: A concrete class having complete implementation of necessary factory methods to instantiate GSPrimeAcc and GSNormalAcc. It extends ShopFactory.

Methods	Access	Return Type	Parameters
getNewPrimeAcc	public	GSPrimeAcc	AccNo, accNm, charges, isSalaried
getNewNormalAcc	public	GSNormalAcc	AccNo, accNm, charges, deliveryCharges

8. The Entry point for application part: Design an entry point for the application to test working of a framework.
    a. Assign instance of GSShopFactory to ShopFactory reference.
    b. Instantiate GSPrimeAcc and refer it through reference PrimeAcc.
    c. Instantiate GSNormalAcc and refer it through reference NormalAcc.
    d. Invoke bookProduct() method.
    e. Invoke toString() method.