Question 1:

You need to create a class named **BankAccount** to handle basic bank‑account operations. The class should have a private field (**balance**) to store the customer’s balance and expose two methods: one for making deposits (**deposit()**) and another for making withdrawals (**withdraw()**). The **deposit()** method must increase the balance, while **withdraw()** must decrease it. The class must block invalid operations—specifically, if the deposit amount is negative or zero, or if the withdrawal amount exceeds the current balance, it should print **“Invalid Option.”**

The class should also include a constructor that initializes the account balance. If the initial balance is negative, it must print **“Invalid Balance”** and set the starting balance to **0.00**. Finally, provide a **get\_balance()** method so users can query the current balance, accurate to the cent.

Test Case:

| **Tests** | **Results** |
| --- | --- |
| BankAccount account1 = new BankAccount(100);  System.out.println(account1.getBalance()); | 100.00 |
| BankAccount account2 = new BankAccount(100);  account2.withdraw(200);  System.out.println(account2.getBalance()); | Invalid Option  100.00 |
| BankAccount account3 = new BankAccount(100);  account3.deposit(50);  System.out.println(account3.getBalance()); | 150.00 |
| BankAccount account4 = new BankAccount(150);  account4.withdraw(30);  System.out.println(account4.getBalance()); | 120.00 |
| BankAccount account5 = new BankAccount(0);  account5.withdraw(50);  System.out.println(account5.getBalance()); | Invalid Option  0.00 |
| BankAccount account6 = new BankAccount(100);  account6.deposit(Integer.MAX\_VALUE);  System.out.println(account6.getBalance()); | 2147483747.00 |
| BankAccount account7 = new BankAccount(120);  account7.withdraw(200);  System.out.println(account7.getBalance()); | Invalid Option  120.00 |
| BankAccount account8 = new BankAccount(120);  account8.deposit(-20);  System.out.println(account8.getBalance()); | Invalid Option  120.00 |