

# ASSIGNMENT-3

1. Write java Program for Consider a scenario, Bank is a class that provides functionality to get rate of interest. But, rate of interest varies according to banks. For example, SBI, ICICI and AXIS banks could provide 8%, 7% and 9% rate of interest.(Method Overriding)

CODING:

```
class Bank {  
    float getRateOfInterest() {  
        return 0.0f;  
    }  
}  
  
class SBI extends Bank {  
    @Override  
    float getRateOfInterest() {  
        return 8.0f;  
    }  
}  
  
class ICICI extends Bank {  
    @Override  
    float getRateOfInterest() {  
        return 7.0f;  
    }  
}  
  
class AXIS extends Bank {  
    @Override  
    float getRateOfInterest() {  
        return 9.0f;  
    }  
}
```

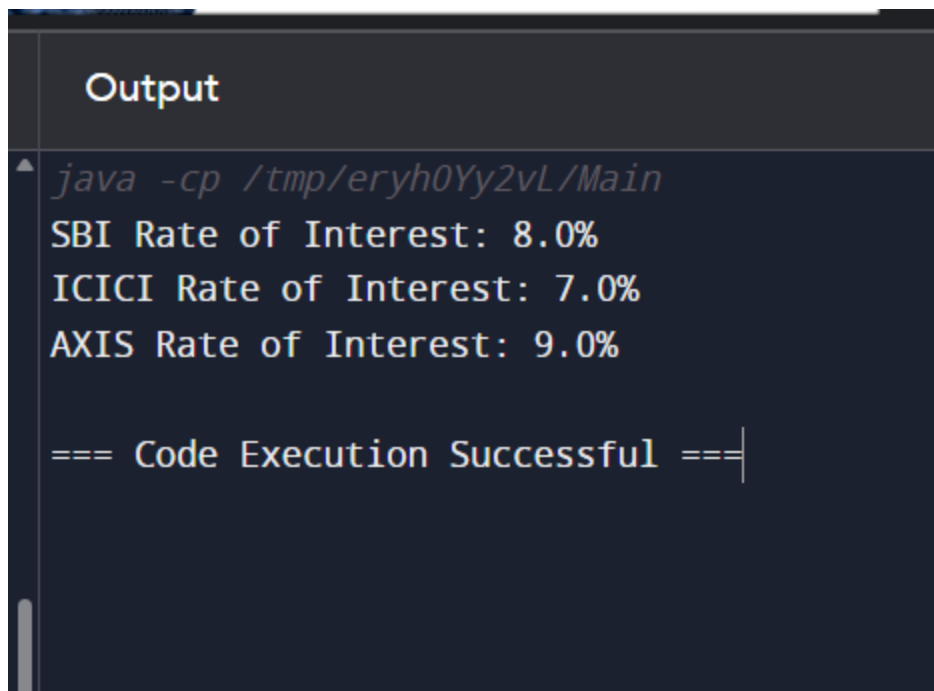
```

    }
}

public class Main {
    public static void main(String[] args) {
        Bank sbi = new SBI();
        Bank icici = new ICICI();
        Bank axis = new AXIS();
        System.out.println("SBI Rate of Interest: " + sbi.getRateOfInterest() + "%");
        System.out.println("ICICI Rate of Interest: " + icici.getRateOfInterest() +
"%");
        System.out.println("AXIS Rate of Interest: " + axis.getRateOfInterest() + "%");
    }
}

```

OUTPUT:



The screenshot shows a dark-themed IDE output window. At the top, the title bar says "Output". Below it, the command `java -cp /tmp/eryh0Yy2vL/Main` is shown in a lighter font. The output consists of three lines: `SBI Rate of Interest: 8.0%`, `ICICI Rate of Interest: 7.0%`, and `AXIS Rate of Interest: 9.0%`. At the bottom, a green message reads `=== Code Execution Successful ===` with a cursor at the end.

2. . Develop a JAVA code to display the balance. Include the following members:

- Design a class to represent a bank account.

•Data Members: Name of the depositor, Account number, Type of account(Savings/Current), Balance amount in the account(Minimum balance is Rs.500.00)

•Methods:

- 1.To read account number, Depositor name, Type of account.
- 2.To deposit an amount (Deposited amount should be added with it)
- 3.To withdraw an amount after checking balance(Minimum balance must be Rs.500.00)

Note : Assume that balance amount = 10000

Test Cases

- 1.100, Raja, S, 8000
- 2.Raja, 100, S, 9000
- 3.101, Rani, S, 12000
- 4.102, Ragu, W, 8000
- 5.103, Ravi, C, 10000

CODING:

```
class BankAccount {  
    private String depositorName;  
    private int accountNumber;  
    private String accountType;  
    private double balance;  
  
    public BankAccount(String depositorName, int accountNumber, String  
accountType, double balance) {  
        this.depositorName = depositorName;  
        this.accountNumber = accountNumber;  
        this.accountType = accountType;  
        this.balance = balance >= 500 ? balance : 500;  
    }  
  
    public void displayAccountDetails() {  
        System.out.println("Account Number: " + accountNumber);  
        System.out.println("Depositor Name: " + depositorName);  
    }  
}
```

```
System.out.println("Account Type: " + accountType);  
System.out.println("Balance: " + balance);  
}
```

```
public void deposit(double amount) {  
    if (amount > 0) {  
        balance += amount;  
        System.out.println("Amount deposited: " + amount);  
        System.out.println("New balance: " + balance);  
    } else {  
        System.out.println("Invalid deposit amount");  
    }  
}
```

```
public void withdraw(double amount) {  
    if (amount > 0 && balance - amount >= 500) {  
        balance -= amount;  
        System.out.println("Amount withdrawn: " + amount);  
        System.out.println("New balance: " + balance);  
    } else if (amount <= 0) {  
        System.out.println("Invalid withdrawal amount");  
    } else {  
        System.out.println("Insufficient balance to maintain minimum balance of  
Rs.500");  
    }  
}
```

```
public static void main(String[] args) {  
    BankAccount account1 = new BankAccount("Raja", 100, "S", 8000);  
    account1.displayAccountDetails();  
    account1.deposit(1000);  
    account1.withdraw(9000);  
}
```

```
BankAccount account2 = new BankAccount("Rani", 101, "S", 12000);
account2.displayAccountDetails();
account2.deposit(2000);
account2.withdraw(5000);
BankAccount account3 = new BankAccount("Ragu", 102, "W", 8000);
account3.displayAccountDetails();
account3.deposit(500);
account3.withdraw(8500);
BankAccount account4 = new BankAccount("Ravi", 103, "C", 10000);
account4.displayAccountDetails();
account4.deposit(500);
account4.withdraw(2000);
}
}
```

OUTPUT:

## Output

```
java -cp /tmp/dzCWrVHIBL/BankAccount
```

Account Number: 100

Depositor Name: Raja

Account Type: S

Balance: 8000.0

Amount deposited: 1000.0

New balance: 9000.0

Insufficient balance to maintain minimum balance of Rs.500

Account Number: 101

Depositor Name: Rani

Account Type: S

Balance: 12000.0

Amount deposited: 2000.0

New balance: 14000.0

Amount withdrawn: 5000.0

New balance: 9000.0

Account Number: 102

=== Code Exited With Errors ===



ENG  
IN

