LAB SET - 3

Assignment-1.

- Create a BankAccount class
- BankAccount class should have three fields accoundHolderName (String), bankName(String), accountBalance(double).
- Create a constructor that takes account holder's name, bank name and initial balance.
- Add three methods to the interface - getBalance(), deposit() and withdraw().
- Implement all three methods.
- In the main method create three bank accounts with different account holders names and ICICI, HDFC and SBI as bank names.
- Deposit and withdraw money for each account. Display the account balance.

Source code

BankAccount.java | Main.java

```java
package anudip.java.lab;

public class Main {
    public static void main(String[] args) {
        // Create three BankAccount objects with different account holders and banks
        BankAccount acc1 = new BankAccount("Harish", "ICICI", 68000);
        BankAccount acc2 = new BankAccount("Dex", "HDFC", 45800);
        BankAccount acc3 = new BankAccount("Suhas", "SBI", 26900);

        // Deposit and withdraw operations on first account
        acc1.deposit(2000);    // Deposit 2000 to Raj's account
        acc1.withdraw(1000);   // Withdraw 1000 from Raj's account
        acc1.printAccountDetails(); // Print Raj's account details

        // Deposit and withdraw operations on second account
        acc2.deposit(2500);
        acc2.withdraw(3000);
        acc2.printAccountDetails();

        // Deposit and withdraw operations on third account
        acc3.deposit(5000);
        acc3.withdraw(4500);
        acc3.printAccountDetails();
    }
}
```

BankAccount.java | Main.java

```java
package anudip.java.lab;

class BankAccount {
    private String accountHolderName;
    private String bankName;
    private double accountBalance;

    // Constructor to initialize the bank account
    public BankAccount(String accountHolderName, String bankName, double initialBalance) {
        this.accountHolderName = accountHolderName;
        this.bankName = bankName;
        this.accountBalance = initialBalance;
    }

    // Method to get the current balance
    public double getBalance() {
        return accountBalance;
    }

    // Method to deposit money into the account
    public void deposit(double amount) {
        if (amount > 0) {
            accountBalance += amount;
            System.out.println(amount + " deposited. New balance: " + accountBalance);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    // Method to withdraw money from the account
    public void withdraw(double amount) {
        if (amount <= 0) {
            System.out.println("Withdrawal amount must be positive.");
        } else if (amount <= accountBalance) {
            accountBalance -= amount;
            System.out.println(amount + " withdrawn. New balance: " + accountBalance);
        } else {
            System.out.println("Insufficient funds.");
        }
    }

    // Method to print account details
    public void printAccountDetails() {
        System.out.println("Account Holder: " + accountHolderName + ", Bank: " + bankName + ", Balance: " + accountBalance);
    }
}
```

Output

Problems | Javadoc | Declaration | Console

```
<terminated> Main [Java Application] C:\Users\Preetham\.p2\pool\plugins\org.eclipse.justj.ope
2000.0 deposited. New balance: 70000.0
1000.0 withdrawn. New balance: 69000.0
Account Holder: Harish, Bank: ICICI, Balance: 69000.0
2500.0 deposited. New balance: 48300.0
3000.0 withdrawn. New balance: 45300.0
Account Holder: Dex, Bank: HDFC, Balance: 45300.0
5000.0 deposited. New balance: 31900.0
4500.0 withdrawn. New balance: 27400.0
Account Holder: Suhas, Bank: SBI, Balance: 27400.0
```

Assignment-2.

● Write a Java program that demonstrates method overriding by creating a superclass called Animal and two subclasses called Dog and Cat.

● The Animal class should have a method called makeSound(), which simply prints "The animal makes a sound."

● The Dog and Cat classes should override this method to print "TheCat/The dog meows/barks" respectively.

● The program should allow the user to create and display objects of each class.

Source code

Animal.java | Main.java

```java
package anudip.java.lab;

class Animal {
    // Method to simulate the sound an animal makes
    public void makeSound() {
        System.out.println("The animal makes a sound.");
    }
}

//Dog class inherits from Animal and overrides the makeSound method
class Dog extends Animal {
    @Override
    public void makeSound() {
        System.out.println("The dog barks.");
    }
}

//Cat class inherits from Animal and overrides the makeSound method
class Cat extends Animal {
    @Override
    public void makeSound() {
        System.out.println("The cat meows.");
    }
}
```

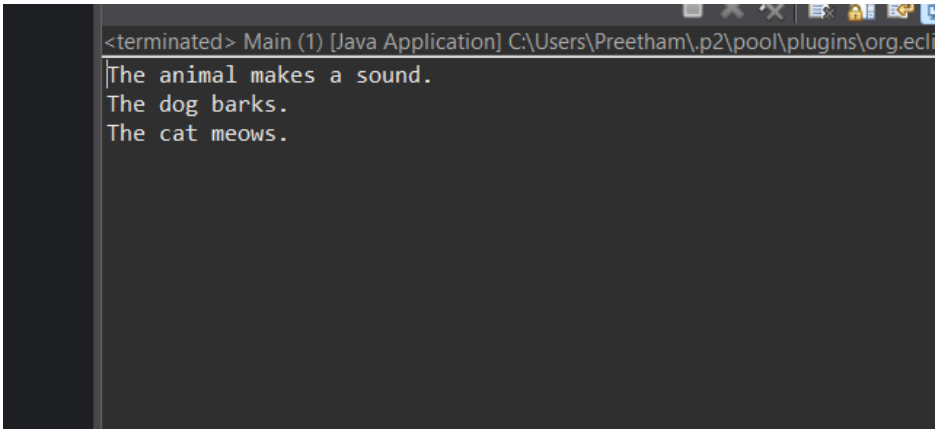
Animal.java Main.java

```java
package anudip.java.lab;

public class Main {
    public static void main(String[] args) {
        // Create an instance of the generic Animal class
        Animal genericAnimal = new Animal();
        // Create an instance of the Dog subclass
        Dog dog = new Dog();
        // Create an instance of the Cat subclass
        Cat cat = new Cat();

        // Call the makeSound method on the generic Animal object
        genericAnimal.makeSound();
        // Call the makeSound method on the Dog object – demonstrates method overriding
        dog.makeSound();
        // Call the makeSound method on the Cat object – demonstrates method overriding
        cat.makeSound();
    }
}
```

Output



Assignment-3.

- Create an enum BankName.

- Create a constructor of the enum which takes a parameter interestRate of type double.

- Create the enum constant -ICICI (6.3), HDFC (5.8), SBI (6.0).

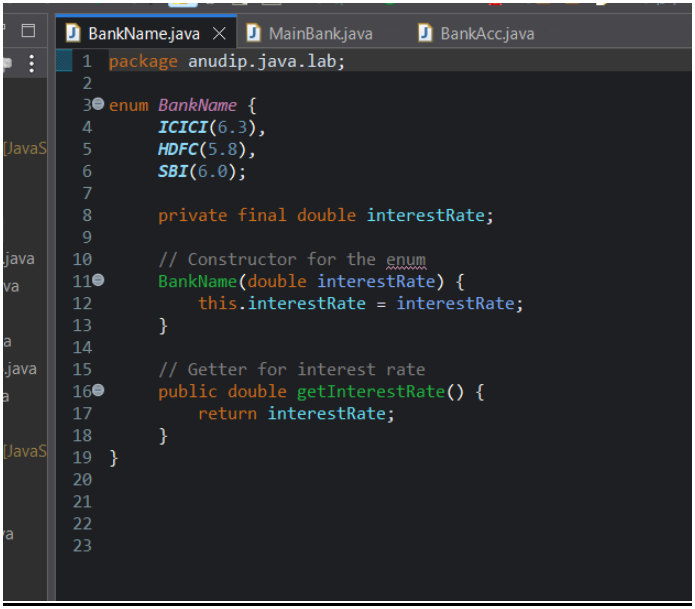
- Now in the BankAccount class of previous exercise, modify the type of bankName to enum BankName. Change constructor accordingly.

- Modify the instantiation of a new Account accordingly. (Example: new BankAccount(12000, "Harry", BankName.SBI);

- Print the bankName for each account.

- Add a method which calculates total interest - calculateInterest (int numberOfYears). The interest will be calculated based on interest rate and available balance. Print total interest. [Hints: double totalInterest = bankName.interestRate* numberOfYears * accountBalance;]

Source code



BankName.java | MainBank.java | BankAcc.java

```java
package anudip.java.lab;

//Class representing a bank account
public class BankAcc{
private double accountBalance;
private String accountHolderName;
private BankName bankName;

// Constructor for BankAccount
public BankAcc(double accountBalance, String accountHolderName, BankName bankName) {
    this.accountBalance = accountBalance;
    this.accountHolderName = accountHolderName;
    this.bankName = bankName;
}

// Method to calculate total interest
public double calculateInterest(int numberOfYears) {
    double totalInterest = bankName.getInterestRate() / 100 * accountBalance * numberOfYears;
    return totalInterest;
}

// Method to print account details
public void printAccountDetails() {
    System.out.println("Account Holder: " + accountHolderName);
    System.out.println("Bank Name: " + bankName);
    System.out.println("Account Balance: " + accountBalance);
}
}
```

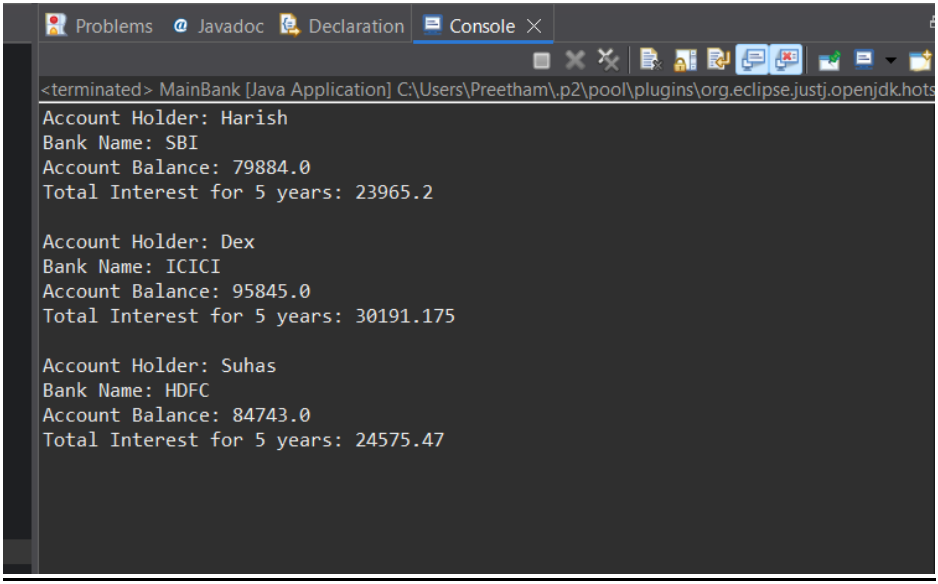
BankName.java | MainBank.java | BankAcc.java

```java
package anudip.java.lab;

//Main class to run the program
public class MainBank {
 public static void main(String[] args) {
      // Creating bank accounts
      BankAcc account1 = new BankAcc(79884, "Harish", BankName.SBI);
      BankAcc account2 = new BankAcc(95845, "Dex", BankName.ICICI);
      BankAcc account3 = new BankAcc(84743, "Suhas", BankName.HDFC);

      // Printing account details and calculating interest
      BankAcc[] accounts = {account1, account2, account3};
      for (BankAcc account : accounts) {
          account.printAccountDetails();
          double interest = account.calculateInterest(5); // Calculate interest for 5 years
          System.out.println("Total Interest for 5 years: " + interest);
          System.out.println();
      }
 }
}
```

Output


Problems
Javadoc
Declaration
Console
<terminated> MainBank [Java Application] C:\Users\Preetham\.p2\pool\plugins\org.eclipse.justj.openjdk.hots
Account Holder: Harish
Bank Name: SBI
Account Balance: 79884.0
Total Interest for 5 years: 23965.2
Account Holder: Dex
Bank Name: ICICI
Account Balance: 95845.0
Total Interest for 5 years: 30191.175
Account Holder: Suhas
Bank Name: HDFC
Account Balance: 84743.0
Total Interest for 5 years: 24575.47