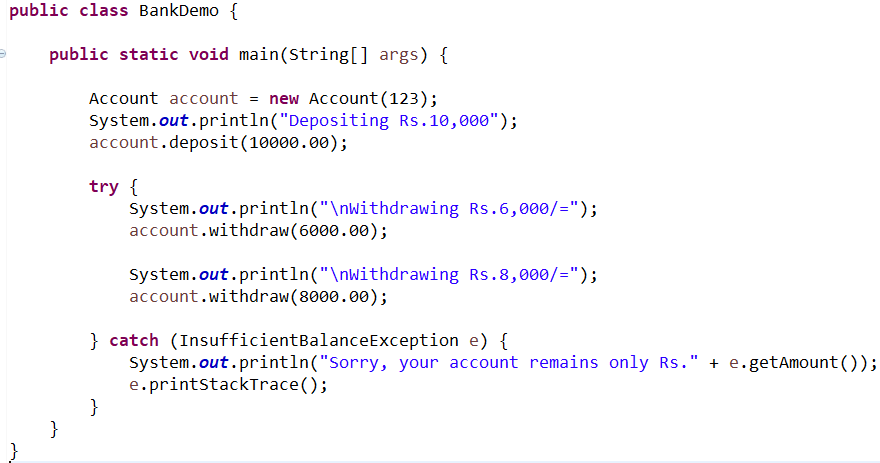
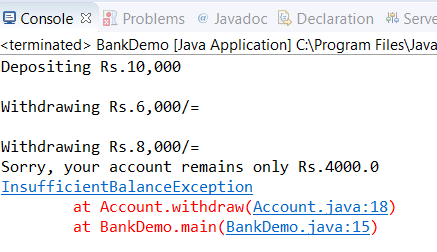
1. Consider the following **BankDemo** Application to perform deposit and withdraw amount from the customer account. To perform these operations, you should create an **Account** class and validate the withdrawal amount lest make the account **overdue.** You should create custom exception class **“InsufficientBalanceException”.**  **[Marks - 20]**

The sample **BankDemo** Application main program is given below with sample output. Your implementation should satisfy the same.



When you withdraw more than the existing account throw **InsufficientBalanceException.** When you run the program out put should be as follows.



1. Create **InsufficientBalanceException** class and amount should be able to pass through the constructor of this custom exception class

**public** **class** InsufficientBalanceException **extends** Exception {

**private** **double** amount;

**public** InsufficientBalanceException(**double** amount) {

**this**.amount = amount;

}

**public** **double** getAmount() {

**return** amount;

}

}

**(05 marks)**

1. Create **Account** class that holds **balance** and **Account No**. Implement operations to display existing balance, account number and account number can be assigned through the Constructor

**public** **class** Account {

**private** **double** balance;

**private** **int** accNo;

**public** Account(**int** accNo) {

**this**.accNo = accNo;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **int** getNumber() {

**return** accNo;

}

}

**(05 marks)**

1. Implement the **deposit** operation and that increases the existing balance in the account

**public** **void** deposit(**double** amount) {

balance += amount;

}

**(02 marks)**

1. Implement the withdraw operation and that reduces the balance with given value. In case if balance is not sufficient **throw InsufficientBalanceException** in the method and you should handle it in the BankDemo Application. You throw this in the withdraw operation as below

**throw new InsufficientBalanceException(amount);**

**public** **void** withdraw(**double** amount) **throws** InsufficientBalanceException {

**if**(amount <= balance) {

balance -= amount;

}**else** {

**double** needs = amount - balance;

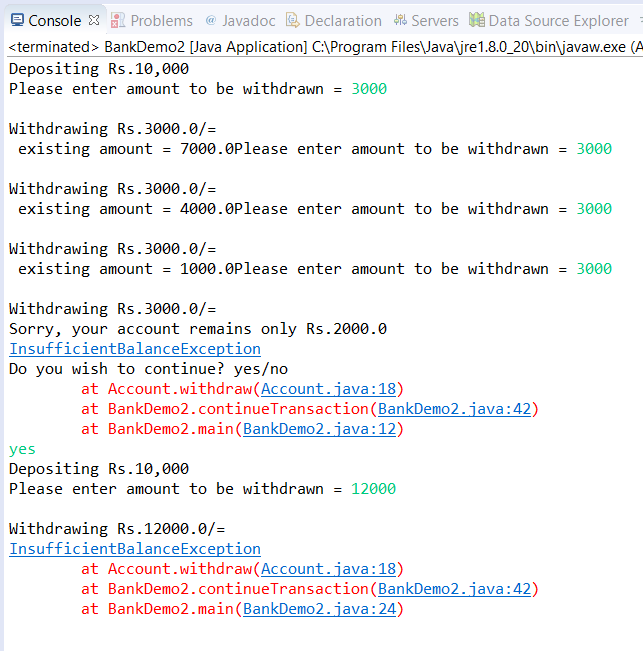
**throw** **new** InsufficientBalanceException(needs);

}

}

**(08 marks)**

1. Modify the above **BankDemo** class to give the below output [**30 marks]**



***(No need to consider the keyboard input validations in your implementation)***

1. In the modified program user should enter the withdrawal amount as keyboard input and this activity should continue as infinite loop until user response for the question “**Do you wish to continue ?”** If user answers as “**no**” program will terminate

**(12 marks)**

1. You should extend the above exception handling with including **finally block**. In the finally block you should ask the above question “**Do you wish to continue?”**

**(08 marks)**

1. If user response “**yes”** for the above question **a)** in your program should deposit the same amount for the account and continue the withdrawal process

**(05 marks)**

1. Make sure you should not duplicate the logics in the program for above modification (Consider OOP concepts)

**(05 marks)**

**public** **static** **void** main(String[] args) {

Account account = **new** Account(123);

System.***out***.println("Depositing Rs.10,000");

account.deposit(10000.00);

**try** {

*continueTransaction*(account);

} **catch** (InsufficientBalanceException e) {

System.***out***.println("Sorry, your account remains only Rs." + e.getAmount());

e.printStackTrace();

} **finally** {

System.***out***.println("Do you wish to continue? yes/no");

Scanner scanner = **new** Scanner(System.***in***);

**if**(scanner.next().equalsIgnoreCase("yes")){

**try** {

System.***out***.println("Depositing Rs.10,000");

account.deposit(10000.00);

*continueTransaction*(account);

} **catch** (InsufficientBalanceException e) {

e.printStackTrace();

}

**else**{

System.***out***.println("Good Bye");

}

}

}

**public** **static** **void** continueTransaction(Account account) **throws** InsufficientBalanceException {

System.***out***.print("Please enter amount to be withdrawn = ");

Scanner scanner = **new** Scanner(System.***in***);

**double** amount;

**while** (**true**) {

amount = scanner.nextDouble();

System.***out***.print("\nWithdrawing Rs." + amount + "/="); account.withdraw(amount);

System.***out***.println(" existing amount = " + account.getBalance());

System.***out***.print("Please enter amount to be withdrawn = ");

}

}

**Total marks = 50**