

Experiment No: 4

Aim: To implement the following

- i) A class "Account" with minimum balance 1000rs, deposit() method to deposit amount, withdraw() method to withdraw amount and also throws LessBalanceException if an account holder tries to withdraw money which makes the balance less than 1000rs.
- ii) A class "LessBalanceException" which returns the statement that says "withdraw amount (___ rs) is not valid".
- iii) A class which creates 2 accounts through which both deposit and withdraw operations are performed. Appropriate action has to be taken for LessBalanceException.

Theory: An exception (or exceptional event) is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program or Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled. Exceptions in java can arise from different kind of situations such as wrong data entered by user, hardware failure, network connection failure, Database server down etc.

Program:

```
package five;
import java.util.*;
class LessBalanceException extends Exception //Defining user-defined exception class
    "LessBalanceException"
{
    LessBalanceException(String s) //Constructor which takes user error message
    {
        super(s); //calls super class constructor
    }
}

class Account //class account
{
```

```

double minbal;
Account()
{
    minbal=1000;    //minimum balance of any account is made 1000rs
}
void deposit(double amt)
{
    minbal=minbal+amt; //deposit amount is added to minimum balance
}
void withdraw(double amt)
{
    double t;
    t=minbal;
    t=t-amt;
    try {
        if (t < 1000) //check against balance not to be lesser than 1000rs at any time
            throw new LessBalanceException("Balance can't go less then 1000 (Withdraw amount is
not
                                valid)");

        else {
            minbal = minbal - amt; //withdraw amount is subtracted from latest minimum balance
            System.out.print("Balance after withdraw:");
            System.out.println(minbal);
        }
    }
    catch(LessBalanceException l)
    {
        System.out.println(l.getMessage());
    }
}

double balance()
{
    return minbal;
}
}

```

```

public class Five {
    public static void main(String args[])    {
        Account a1=new Account(); //creating account1
        Account a2=new Account(); //creating account2
        double dep,wit;
        Scanner sc=new Scanner(System.in);
        System.out.print("Min bal of account1: ");
        System.out.println(a1.balance()); //account1 minimum balance
        System.out.println("Enter the amount to deposit: ");
        dep = sc.nextDouble();
        a1.deposit(dep); //account1 deposit method call
        System.out.print("Balance after deposit:");
        System.out.println(a1.balance());
        System.out.println("Enter the amount to Withdraw: ");
        wit = sc.nextDouble();
        a1.withdraw(wit); //account1 withdraw method call
        System.out.print("\nMin bal of account2: ");
        System.out.println(a2.balance()); //account2 minimum balance
        System.out.println("Enter the amount to deposit: ");
        dep = sc.nextDouble();
        a2.deposit(dep); //account2 deposit method call
        System.out.print("Balance after deposit:");
        System.out.println(a2.balance());
        System.out.println("Enter the amount to Withdraw: ");
        wit = sc.nextDouble();
        a2.withdraw(wit); //account2 withdraw method call
    }
}

```