Module: Core Java

Session 13: Exception Handling Practice

* This is a practice session; you will work on Exception handling assignments.
* You can discuss your doubts with the trainer

**Assignments:**

**Assignment 1 – Exception handling using custom exception**

Assume you are an employee in an organisation and use Online Leave Management System if you have to apply for leave. This system sanctions leaves if you have leaves in your quota. Otherwise it should generate an exception called LeaveQuotaExceededException. Assume that you have 20 leaves which the system can grant. Write a java application which has the following:

a) An exception class called LeaveQuotaExceededException

b) A class called LeaveSystem which has

i) A constructor which takes the total leaves

ii) A method which checks if you exceed the leaves and generates an exception if you exceed leaves otherwise sanctions a leave and reduces the same in system

c) An executable class with main method.

**Assignment 2 – Banking application with custom exception handling**

Create the following classes and complete the application:

File Name: InsufficientFundsException.java

You just need to extend the Exception class to create your own Exception class. These are considered to be checked exceptions: Class name: InsufficientFundsException Instance variables: amount:double Instance methods: getAmount():double

File Name: CheckingAccount.java

To demonstrate using our user-defined exception, the CheckingAccount class should contains a withdraw method that throws an InsufficientFundsException: Class Name: CheckingAccount Instance Variables: accountNo:integer balance: double Instance Method 1: checkAccount(accountNo):Boolean Instance Method 2: deposit(amount):void Instance Method 3: withdraw(amount):double

File Name: BankDemo.java

The BankDemo program demonstrates invoking the deposit() and withdraw() methods of CheckingAccount.

You have to treat this class as main class. In this you have to create

CheckAccount object by passing the account number.

You can deposit the amount into that account by invoking the deposit() method by passing the deposit amount as an argument, only, if account exists.

If you have balance in your account, you have to withdraw the amount by invoking the withdraw() method by passing suitable amount as an argument. But, if your requested amount is more than the available balance, you have to generate “InsufficientFundException” object and the same have to be thrown to the caller (with suitable message).

**Assignment 3 – Caselet**

The following program is a user-defined exception called InsufficientFundsException which generates an exception when balance is less than the withdrawn amount. Complete the below caselet to make it work.

package com.ts.exceps; //make this class an exception class

class InsufficientFundsException {

String message;

public InsufficientFundsException(String message){

super();

this.message = message;

}

}

class Bank{

double balance;

public Bank(double balance){

this.balance = balance;

}

//handle the exception

public void withdraw(double amountwithdrawn)

{

if(balance > amountwithdrawn)

balance = balance - amountwithdrawn;

else

//write code to call the InsufficientFundsException here

}

}

public class Test{

public static void main(String args[]) throws InsufficientFundsException{

Bank bank = new Bank(1000.00);

bank.withdraw(2000.00);

}

}