S.Rajalakshmi | B.Senthil Kumar | S. Lakshmi Priya
SSN COLLEGE OF ENGINEERING
Department of Computer Science & Engineering
UCS1313: Object Oriented Programming Using Java Lab
2019-2020 Odd – III Semester
Assignment – V: Packages & Exception Handling
==

Objective:

1. To create and use the packages

2. To throw the pre-defined and custom-defined exceptions.

----

1. Create a package named `myPack.CurrConvertor` to implement the currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa). Write a java program which uses this package to convert the given currency into other currency type.

Use the pre-defined exception `java.lang.NumberFormatException` (checked) to throw the error whenever the given input currency is not in the required format.

2. Create a class named “Account” which contains customer name, acct_num, branch, balance, PAN. Consider the situation where a customer holds only one account per branch and there are many accounts owned by different customers in the same branch. The following constraints are identified:

a. Write a function: `Account Search(Account []acc)` that checks for a particular acct_num. If not present then the exception “`AccountNotFoundException`” should be handled by the calling function.

b. In deposit, if the customer deposits amount more than Rs. 50,000 in case of account not having PAN number, then cause the exception “`PANRequiredException`”.

c. While during withdrawal, check for the following constraints:
i) If the withdrawal is more than the balance then “`NoEnoughMoneyException`”.
ii) Check for the minimum balance before the withdrawal such that: minimum balance > balance - withdrawal. Else rise the exception: “`MinBalRequiredException`”.

Custom-defined Exception:

When defining a custom-defined exception, use the `String toString()` method to display the custom defined error messages (description) instead of regular error format – `exception object:error description`.

Design a class called Account as described below:

Account
-cname:String -pan:int -accno:int -branch:String -balance:float
+Account(cname,accno,branch,balance) +Account(cname,pan,accno,branch,balance) +getMethod(): all – fields +setPan(panno):void +setBalance(amt):void +deposit(accno, amt):void +withdraw(accno, amt):void **+toString():String**

############$$$$$$$$$$$$$$$$$#####################$$$$$$$$$$$$$$$####