Object oriented programming
LAB - VII

1. Write a Java program that reads a numeric input from the user. Call the input method in a try block and catch the specific exception that the input method would throw if the user entered a non-numeric input.
Hint : import java.util.InputMismatchException;

2. Case study : Bank management system
Define a package bank.
Include following functionalities as abstract methods in an interface
abstract XXX credentialsCheck(String Username, String password);
abstract XXX credit(XXX);
abstract XXX debit(XXX);
abstract XXX displayBalance(XXX);
abstract XXX exit();

Until user selects exit he can perform any of the above operations.
Handle the following exceptions during the transaction

1. Username and password mismatch. // display error message and continue;
2. If debit amount exceeds available balance. // display error message and continue;

3.Case study : Tax fraud detection
Define package audit.
Include following functionalities as abstract methods in an interface.
abstract XXX taxChecker (XXX);
abstract XXX taxPaid(XXX);
abstract XXX homeExpenditure(XXX);
abstract XXX healthExpenditure(XXX);
abstract XXX vehicleExpenditure(XXX);
abstract XXX personalFamilyExpenditure(XXX);
abstract XXX miscellaneousExpenditure(XXX);

Inputs : all the above expenditures, TotalIncome and Taxpaid
He/She has to pay 10% of (TotalIncome- (all other expenditures)) on mismatch throw an exception as fraud and display message, how much he/she has to pay.

4. Mr.Bean claims himself as efficient software developer and claims that his application can handle the following exceptions
ArithmeticException, NullPointerExceptions, ArrayIndexOutOfBoundsException, IOException etc., . You being the Tester test the application to validate his claims.
Hint : Develop any application that handle's all the exceptions stated above.

5. Passengers can reserve their tickets in train. It has two First class A/C coaches (B1, B2), two sleeper coaches(s1, s2) with capacity 70 each.
Prompt the user to enter the required berths and class(A/c or sleeper), allocate berths randomly and display the Confirmed berths after booking.

Under following conditions throw an exception as "You may be an Agent" and deny the request.

1. If the number of required berths are above 6.
2. If he/she is booking tickets by selecting different classes (A/c or sleeper).