1. Write a program to throw a checked exception explicitly using 'throw' keyword and

a) Handle the exception in same method.
b) use throws clause and handle the exception in some other method (calling method)
c) Don't either handle or use the throws clause.

2. Repeat program 2 with unchecked Exception and demonstrate the difference in both program.

3.Create a user defined exception to check whether your employee exist in your data structure (use any data structure to store the employees - like array, ArrayList etc) and throw exception if name is not in the employees list. Use the catch and finally block to make an appropriate handling.

4.The assignment requirements follow:

a. Create your exception in a file called ScoreException.java. Create the UseScoreException.java class to use this exception. The UseScoreException.java file will have the main() method.

b. Prompt the user to enter a test score. Read this score using a dialog box.

c. Any input value greater than 100 or less than 0 should generate the ScoreException exception. You must throw a ScoreException exception at least once and you must have a catch block that catches a ScoreException exception. Use the getMessage method of the Throwable class to display the message associated with this exception. You may do other processing also if you wish. This catch block must display a dialog box with "The score must be >= 0 and <= 100!".

d. Display a return message with "Do you want to enter another score?" after the user inputs 'Yes' or 'Y'.

e. Use the "parseXXXX" methods to convert the String read from the input score. This method will throw a NumberFormatException if it is unable to convert the String object to a valid number. Any input value that has characters other than numbers, a decimal, or a negative sign will generate the NumberFormatException exception. You must have a catch block that catches a NumberFormatException exception. This catch block must display a dialog box with "You must enter a number for the score!".

f. Return step d.

g. Display "That is a valid score." if a valid score is entered..

h. Continue the steps above until the user enters No. ("Do you want to enter another score?").