# JAVA Programming Assignment
## Exception Handling

1. Write a java program for following. Ask user to input the size of a one-dimensional integer array at run-time. Also ask user to input a series of non-zero, integer values and store those values in defined one-dimensional array. Write code to handle exception for following conditions:
   a. Create an exception handler for values that are not in an integer format (if user enters value except integer). Repeatedly ask user to input value until user enter correct integer format value.
   b. Create an exception handler for the array index being out of bounds (User try to print the value at array index which does not exist).
   c. Create an exception handler for an arithmetic exception.
   d. Calculate and output the arithmetic mean of the numbers in the array.

2. Write a program that prompts the user to enter the length in feet and inches and outputs the equivalent length in centimeters. If the user enters a negative number, nondigit number, or numbers not in correct feet, inch format, throw and handle an appropriate exception and prompt the user to enter another set of numbers.

3. Write a program that prompts the user to enter time in 12-hour notation. The program then outputs the time in 24-hour notation. Your program must contain three user defined exception classes: InvalidHrExcep, InvalidMinExcep, and InvalidSecExcep. If the user enters an invalid value for hours, then the program should throw and catch an InvalidHr object. Similar conventions for the values of minutes and seconds.

4. Write a Program that gets the input from user in forms of String and Print them in the file. Then from the file it will read all the character and find out the one that has highest frequency. Make use of Character Streams.

5. Write a Program that has a student class with members Enroll_no, Name, Percentage. In the public static void main(), create the two objects of Student class and write them in the file. Later on open the file in read mode and retrieve the details of Students in form of Student Objects. Make use of ObjectInputStream and ObjectOutputStream. (Remember that the student class must implement Serializable Interface to store and retrieve the objects to/from the file)