PROGRAMMING EXERCISES(FILE I/O STREAMS AND THREADS)

1. Text 1= “Today is Saturday and it is a gloomy rainy day. I am the student of Object-Oriented Programming”

   Text 2= “we need help”?

   Write a program that should create a text file with a name OOPs in D:\ drive (or in your local hard drive) and store the above mention text of text 1 in file then add above mention text of Text 2 in the same file in reverse order.

2. Write a Java application that will be able to add, subtract, multiply, divide, compare, convert to floating point, reduce to its lowest terms, and find absolute value for rational numbers. Your program should be written in Object Oriented Programming style. It should read a file called **input.txt** that contains a pair of rational numbers to a line.

   (**NOTE**: the line may contain incorrect input). It should parse each line to retrieve two rational numbers and perform operations on them. Rational numbers will be separated by “and”. The results must be written to a file called **results.txt**.)

3. Create an example of a “busy wait.” One thread sleeps for awhile and then sets a flag to **true**. The second thread watches that flag inside a while loop (this is the “busy wait”) and when the flag becomes **true**, sets it back to false and reports the change to the console. Note how much wasted time the program spends inside the “busy wait” and create a second version of the program that uses **wait( )** instead of the “busy wait.”

4. Write a program that runs 5 threads, each thread randomizes a number between 1 and 10. The main thread waits for all the others to finish, calculates the sum of the numbers which were randomized and prints that sum. You will need to implement a *Runnable* class that randomizes a number and store it in a member field. When all the threads have done, your main program can go over all the objects and check the stored values in each object.