Write a Java program to read 5 subject marks of a student and calculate the total and grade. The grade system is as follows.

Letter Grade	**Grade Points**	**Marks Range**
O (Outstanding)	10	91 – 100
A+ (Excellent)	9	81 – 90
A (Very Good)	8	71 – 80
B+ (Good)	7	61 – 70
B (Average)	6	50 – 60
RA	0	< 50

a. Define a class named **COMPLEX** for representing complex numbers that contains necessary data members and member functions. A complex number has the general form a + ib, where a is the real part and b is the imaginary part (i stands for imaginary). Include methods for all the four basic arithmetic operators.

b. Write a Java program that determines the number of days in a month.

Write a Java program to create a Package "YEAR_I" which has a class YearIMarks (members – sub1mark, sub2mark). Create another package "YEAR_II" which has a class YearIIMarks (members – sub3mark, sub4mark). Create n objects of Student class (having rollNumber, name, YearIMarks and YearIIMarks). Calculate the Grade ('Pass' > =50 else 'Fail') for each subject and display the result of the student in proper format.

Create a package named 'com'. Define subpackages;

- 'transact': with class 'Transaction' with static methods credit() and debit()
- 'loan': with class 'LoanAccount' with method doTransaction() which calls Transaction class methods.

Create one 'LoanAccount' object in main to perform operations on it by accepting command line arguments.

Define an interface "QueueOperations" which declares methods for a static queue. Define a class "MyQueue" which contains an array and front and rear as data members and implements the above interface. Initialize the queue using a constructor. Write the code to perform operations on a queue object.

Write a java class called 'student' with name, and rollno. Write a class 'Result' to get Marks of 3 subjects and another class "Sports' to get the points obtained in sports. Calculate the total Marks and displays the result (pass or fail) with points obtained in sports for three students using inheritance and constructor.

Define an abstract class "car" with members reg_no, model, reg_date. Define two subclasses of this class – "transportVehicles " (validity_no, start_date, period) and "privateVehicle " (owner_name, owner_address). Define appropriate constructors. Create n objects which could be of either transportVehicles or privateVehicle class by asking the user's choice. Display details of all "privateVehicle" objects and all "transportVehicles" objects.

Create an interface "CreditCardInterface" with methods to viewCreditAmount, viewPin, changePin and payBalance. Create a class Customer (name, card number, pin, creditAmount – initialized to 0). Implement methods of the interface "CreditCardInterface" in Customer class.

Create an array of customer objects and perform the following actions.

- Pay Balance
- Change Pin

Write a Java program to perform the following task.

- Take an integer array of size 20, initialize values randomly between 10 and 90, simultaneously sum all values and calculate average. Now separate values below average and above average in ArrayLists. Finally print both lists in 2 separate rows.

a. Write a java program that reads a string from inputs containing first name, last name and computes an e-mail address with first 3 letters of the first name, first 4 letters of last name, '.' separator and domain. Display the outputs by invoking objects.
b. Create a java abstract class to implement stack concept. Check for the overflow and empty conditions

Write a java program for exception handling:

a. To create a user defined exception whenever user input the word "hello".
b. To add two integers and raise exception when any other character except number (0 – 9) is given as input.

Create a class Doctor with attributes id, name, age and department. Initialize values through parameterized constructor. If age of Doctor is not in between 25 and 65 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.

A program accepts two integers as command line arguments. It displays all prime numbers between these two. Validate the input for the following criteria: Both should be positive integers. The second should be larger than the first. Create user defined exceptions for both.

Write a Java program 'WordCount' that counts the words in one or more files. Start a new thread for each file. For example, if you call

"java WordCount report.txt address.txt Homework.java "

then the program might print

address.txt: 1052
Homework.java: 445
report.txt: 2099

Write a Java program 'LineCounts.java' that will count the number of lines in each files that is specified on the command line. Note that multiple files can be specified, as in

"java LineCounts file1.txt file2.txt file3.txt".

Write a program TestThreadMany.java that takes a positive integer n from the command line and creates exactly n threads that print out their own name. Here is a sample execution:

$ java TestThreadMany 4

Hello, I am Thread #1
Hello, I am Thread #2

Hello, I am Thread #3
Hello, I am Thread #4

a. Write a Java program to demonstrate that as a high-priority thread executes, it will delay the execution of all lower-priority threads

b. Write a Java program to read from an input file and convert the words to lower case and write it in another file.

Write java programs that include generic method to satisfy the following property.

a. To counts the number of odd integers in an integer list
b. To exchange the positions of two different elements in an array.
c. To find the maximal element in the range [begin, end] of a list.

Create a new Java GUI application to convert miles to kilometers when pressing the "Convert!" button. Note that you need to implement the ActionListener interface and override the actionPerformed() method. Note that 1 mile is equal to 1.609 kilometers.

Create a new Java GUI application that moves a snowman on a panel. The position of the snowman can be changed using the control buttons. With the four buttons, a user can move the snowman to the left, right, up or down position.