Question No.1) Construct a UML class diagram showing the structure of a Course Management System having classes: Course, Library, Book, Person, Student, and Staff. A person can only be a Student or a Staff. Include the relationships, instance variables and methods to enable the names, addresses, and courses of students and staff, the name, phone number, and books in the Library, course code, course name and associated book for the Course, and book title for the Book, wherein students pay an annual fee, each course has a registration fee, the library has a membership fee, books have a purchase price, and staff members are paid a salary. The system should manage the finances through a common interface AccountSystem.

Question No.2) Write Java code for the following considering the data members and relationship details from Question 01. Implement the methods as suggested by the names:

a) class `Person` has the constructor `public Person(String name, String address)`.

b) class Staff: constructor `public Staff(String staffName, double salary)`, method `addCourse(String courseCode, String courseName)` that can add a maximum of 3 courses for a member of staff, and `removeCourse(Course course)` that can remove a course from the staff's list of courses. Use method overriding to return the salary of the staff member.

c) class Student: constructor `public Student(String studentName, double tutionFee)`, methods `addMarks(Course course, int marks)`, that stores the marks of the student for each course in key value pairs, `addCourse(String name, String code, double fee)` that adds a course to the student's list of courses, and an overridden method to return the tuition fee.

d) class `Course:` constructor `public Course(String courseCode, String courseName, Book associatedBook).` The initial registration fee is PKR3000. Use method overriding to return the fee. The class should be able to initialize the course objects for both the Student and Staff classes through their `addCourse()` methods.

e) class `Book` with constructor `public Book(String title, double price).` The class implements an overridden method to return the purchase price of the book.

f) Write a test class showing how each method in the classes is tested.

g) In the test class, polymorphically display the objects for all the classes, including the Library class. Assume there is a toString() method implemented in each class to return its details. Use downcasting to turn the price for Book to 0, displaying a message "`Books are Donated!`".

Question No.3) Write a Validation class that uses static Boolean methods to validate the values for the String data members name, email address, phone number, and registration number, using Regular Expressions, such that a name can only contain English alphabets and the first letter is always a capital letter. The username in an email address cannot start with a non-alphabetic character and may contain non-consecutive underscore or dot characters, while the domain name should match kust.edu.pk. The phone number should have a format like 0-000-0000000 where the first digit is always a 0 and the second is always 3. The pattern for registration number should match the registration number of an SE student of IoC KUST.

Question No.4) Write an application that reads text from a document named mydocument.txt, tokenizes the line using space characters as delimiters and prints out the individual words.

Question No.5) Given a String sentence = "Hello how are you". Write code using String methods that:

a) Prints the words beginning with the letter "h", and the words ending with the letter "e".

(Hint: split the string and store it in an array)

b) Outputs the text in all lowercase letters. Prints index of the word "hello" in the text. Replaces every **x** in the text with an **e.** Prints the character at index 13.