PART B (80 mark)

1.
   a. Describe the role of design patterns in the design and programming process. What benefits do you hope to gain from using design patterns?
   b. When designing a software system what are the common design mistakes designers will do?
   c. What are the essential elements of design patterns?
   d. Compare and contrast design patterns over frameworks.

2. You have implemented a very complex component that takes some input, does some calculations and saves the results into a text file. You want to sell the component, but first you want to restrict the access to the results by saving the contents in the result text file using some encryption algorithm.

   a. What design pattern(s) can be used here to make the component transparent only to the users that have bought the encryption key from you?
   b. Discuss the advantages of using the selected design patterns.
   c. Sketch the UML diagram of the entire system with the design patterns you have chosen.
   d. Suppose you want to use different encryption algorithms. What design pattern(s) can be used to vary the encryption algorithm easily? Justify your answer.

3.
   a. Suppose you want to develop a class with an expected interface which provides some services to your clients. Later on you realize that there is an existing class performs the same services your client needs with different method names. Which pattern would you like to use here and why?

   b. Suppose you are designing the file system of an operating system.
      i. The file system has tree-structured directories. Which design pattern is indicated by the tree structured directories?
      ii. Suppose your system should be able to implement many kinds of file systems. A Linux file system has Linux directories and Linux files, while a BSD file system and NT file system have their own ways of representing directories and files. You want to write reusable algorithms for the file system, but when these algorithms are creating directories and files, they will create them for Linux, BSD or NT as appropriate. Which design pattern would help you create objects of the right class? Justify.