Name : ....

Roll No. : ....

Invigilator's Signature : ....

CS/M.Tech(CSE)/SEM-2/CSEM-204/2012

2012

SOFTWARE ENGINEERING

Time Allotted : 3 Hours *Full Marks* : 70

The figures in the margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

Answer *all* questions from **Section-A** and any *two* from **Section-B**.

SECTION - A

1. a) With respect to the software design principles suggested by Davis, explain what do you understand by the following :
   i) The design process should not suffer from "tunnel vision".
   ii) The design should be traceable to the analysis model.
   iii) The design should not reinvent the wheel.
   iv) Design is not coding, coding is not design.

   b) According to Robert Martin what are the important characteristics of a bad design ? 8 + 2

2. a) What are coupling and cohesion ? What is functional independence ? Why is it considered as a key to good design ? What do you understand by clean decomposition of a design problem into modules ? What is meant by the visibility of a module ? 5

   b) Explain what you understand by the following :
   Procedural cohesion
   Sequential cohesion
   Temporal cohesion
   Content coupling
   Stamp coupling. 5

3. With suitable examples explain the following : 4 × 5

   Interface segregation principle

   Liskov's substitution principle

   Open-Close principle

   Dependency inversion principle.

SECTION - B

Answer any *two* questions.

4. Take into consideration a framework for desktop applications. Such applications are meant to work with documents. A framework for desktop applications contains definitions for operations such as opening. Creating and saving a document. The basic classes are abstract ones, named Application and Document, their clients having to create subclasses from them in order to define their own applications. For generating a drawing application, for example, they need to define the DrawingApplication and DrawingDocument classes. The Application class has the task of managing the documents, taking action at the request of the client (for example, when the user selects the open or save command form the menu). Because the Document class that needs to be instantiated is specific to the application, the Application class does not know it in advance, so it doesn't know what to instantiate, but it does know when to instantiate it. The framework needs to instantiate a certain class, but it only knows abstract classes that can't be instantiated.

   Use Factory Method design pattern to solve the problem by putting all the information related to the class that needs to be instantiated into an object and using them outside the framework. 15

5. Use abstract factory for a GUI framework which should support several look and feel themes, such as Motif and Windows look. Each style defines different looks and behaviours for each type of control : Buttons and Edit Boxes. In order to avoid the hard coding it for each type of control, define an abstract class LookAndFeel which will instantiate, depending on a configuration parameter in the application, one of the concrete factories : WindowsLookAndFeel or MotifLookAndFeel. Each request for a new object will be delegated to the instantiated concrete factory which will return the controls with the specific flavour. 15

6. Let's assume that we design an application with a factory to generate new objects (Account, Customer, Site, Address objects) with their ids, in a multithreading environment. If the factory is instantiated twice in 2 different threads then it is possible to have 2 overlapping ids for 2 different objects. Implement the Factory as a singleton, in correspondence with the open close principle. 15

7. Consider a restaurant where a waiter has to provide a meal menu as follows :

Base items	***Price***	***Additives***	***Price***
Bread	Rs. 5	Milk	Rs. 10
Plain rice	Rs. 10	Egg curry	Rs. 20
Biryani rice	Rs. 20	Chicken curry	Rs. 50
Plain roti	Rs. 2	Matar paneer	Rs. 35
Tandoori roti	Rs. 3	Chana masala	Rs. 30
Chapatti	Rs. 5	Fish fry	Rs. 50

The Base items are implemented using array whereas Additives are implemented using arraylist.

Use a suitable design pattern, enabling the waiter to display the menu along with their price without knowing the implementation details of Base items and Additives. The waiter is also able to display the vegetarian menu if required.

The customer may place an order by selecting at least one base item along with the desired number of additives. Create a billing system using suitable design patterns to provide the customer with a bill containing the description of the items consumed along with the payable amount.

The design need to be extensively enough to support new types of menu items. 15