

[No. of Printed Pages – 6]

CSE431

Enrol. No

[ET]

END SEMESTER EXAMINATION : NOV.-DEC., 2018

OBJECT ORIENTED SYSTEM DESIGN

Time : 3 Hrs.

Maximum Marks : 70

Note: Attempt questions from all sections as directed.

SECTION – A (30 Marks)

Attempt any five questions out of six.

Each question carries 06 marks.

1. Describe various parts of OMT modeling.
2. Explain Case study of POS System.
3. Three entity classes are used in a collaboration – CarSharer, Journey and Address. Each of these classes will be implemented by a (.java) source file. These classes are used across a number of use cases and are grouped together into a CarSharing component as Java (.class) files, here we are just dealing with the source files. There are two other classes MCSUserInterface and MCSControl. Each of these will be implemented

by a (.java) file. The MCSControl component has a dependency on the CarSharing component and on the MCSUserInterface component.

Draw a component diagram showing the source code dependencies. The .class files are grouped together into two Java archive (.jar) files. The MCSControl.class component will need to read a configuration file (MCS.ini) and display a help file (MCS.hlp) when required. The MCSControl (.class) file also has dependencies on the MCSUserInterface (.java) file and the CarSharing (.jar) components.

4. Create a UML Activity Diagram for processing an order. Once an order has been finalized, four parties are involved in processing it: Online Sales, Accounting, Shipping, and Printing. Online Sales sends the order to Printing, where the associated PDF file is inspected. If the file is not suitable, a new file is requested from the customer. Once the file is suitable, Accounting is informed to charge the credit card. While this is done, Printing carries out the actual printing, and sends the result to Shipping. When Shipping has both received the printed products and confirmation from Accounting that payment was successful, the products are shipped to the customer.

5. Consider the following required functionality of a print-on-demand service: The print-on-demand service provides customers the possibility to print posters, flyers, or books on demand. The customer should be able to select a type of product (poster, flyer, or book), a desired quantity, and a paper type. In case a book has to be printed, additionally the customer can choose between hard cover and soft cover. Finally, the customer needs to provide a PDF file containing the desired content. In order for the customer to be able to place an order, he or she must have an account. The customer can create an account by choosing a username/password combination. Furthermore, his or her address and credit card number can be linked to the account, which is required information when placing an order. Once a customer has provided the information for an order, the system checks if all required information is there, either given in the order (type of product, quantity, etc.), or in the account (address and payment information). If any information is lacking, the system will inform the customer that it needs to be added before the order can be placed. Once all information is in place, the order is placed, and the credit card information is sent to the bank for approval. If the bank approves the card, the order is finalized. A printing agent is in charge of actually performing the printing. He or she inspects the provided PDF files of finalized orders. If a file does not meet the quality

P.T.O.

(2121)

requirements, the customer will be informed about this, and the order is temporarily put on hold until the customer has provided a new PDF file. Finally, the administrator monitors if at all times, sufficient paper and ink stock is present. Whenever the amount of paper or ink is running low, an order must be placed at the appropriate supplier (either the paper or ink supplier).

Create a UML Use Case Diagram for the print-on-demand service.

6. Differentiate between the following :

(a) Patterns and frameworks (3)

(b) Coupling and cohesion (3)

SECTION - B (20 Marks)

Attempt any two questions out of three.

Each question carries 10 marks.

7. What do you mean by unified process? How it is different from waterfall model? Describe various steps involved in the Unified Process.
8. Draw use case diagram and state transition diagram for the given scenario.

Suppose we want to develop software for an alarm clock. The clock shows the time of day. Using buttons, the user can set the hours and minutes fields individually, and choose between 12 and 24-hour display. It is possible to set one or two alarms. When an alarm fires, it will sound some noise. The user can turn it off, or choose to 'snooze'. If the user does not respond at all, the alarm will turn off itself after 2 minutes. 'Snoozing' means to turn off the sound, but the alarm will fire again after some minutes of delay. This 'snoozing time' is pre-adjustable.

9. How will you apply GoF design patterns? Explain the usability of adaptor, singleton and factory patterns.

SECTION – C **(20 Marks)**
(Compulsory)

10. (a) Consider the following description of a university where instructors take courses in which students can enroll.

An instructor has a name, address, phone number, email address, and Salary. A student has also a name, address, phone number, email id, mobile no, average marks. A course has a name and a number. When a student is enrolled in a course, the marks for this enrollment are recorded and the current

average as well as the final mark (if there is one) can be obtained from the enrollment. From a student one can obtain a list of courses he or she is enrolled in. Instructors teach courses. Each course has at least one and at most three instructors. There are two types of course: bachelor and master. From a bachelor course students cannot withdraw. From a master course they can.

Draw a class diagram for this university. Add all necessary attributes and methods with visibility modifiers (public, private, etc.). (10)

(b) Write notes on :

(i) GRASP

(ii) Generalization and Aggregation (10)