

APJ Abdul Kalam Technological University
First Semester M.Tech Degree Examination January 2016

Ernakulam II Cluster

COMPUTER SCIENCE AND ENGINEERING

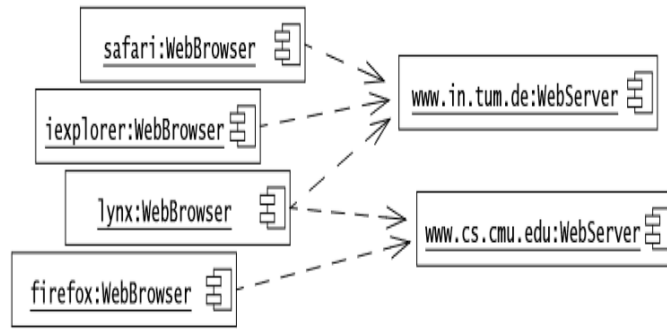
Specialization: COMPUTER SCIENCE AND ENGINEERING

05CS 6007- OBJECT ORIENTED SOFTWARE ENGINEERING

Time: 3 hrs.

Max. Marks: 60

- I. a) Suppose the requirement specification for the Blood Bank Website Development of the blood bank project has been carefully analyzed and the following estimates have been obtained. There is a need for 11 inputs, 11 outputs, 7 inquiries, 22 files, and 6 external interfaces. Also, assume outputs, queries, files function point attributes are of low complexity and all other function points attributes are of medium complexity.
- The complexity adjustment value for factor 1 is set to 3 because the SRS requires that the software product has only a good degree of data communication; factor 2 is set to 0 because the SRS emphasizes no need for heavy use configuration; factor 5 is set to 3 because the order-web-based order fulfillment module has medium level of complex processing; factors 10 and 11 are set to 4 and 2 respectively because the module is always on-line but needs only few updates ; factor 3, 4, 6,7,8,9,12,13,14 are set to 4, 3,2, 3,4,3,4,3,2 respectively based on their estimated level of complexity or demand.
- Make the following calculations showing the full procedure in details:
- i. Calculate the FUNCTION POINTS (FP) for the blood bank project. (5 Marks)
 - ii. Calculate the approximate number of LOC in the following languages: (1 Mark)
 - “C++” programming language
 - “Java” Programming language
 - iii. Calculate the effort and duration assuming that Java is used as the implementation language using basic COCOMO if the development mode is organic. (2 Marks)
- b) How does the risk factor affect the spiral model of software development? (2 Marks)
- c) Can a role be shared between two or more participants? Why or why not? (2 Marks)
- II. Explain requirement elicitation? Discuss the products of requirement elicitation and analysis? (12 marks)
- III. a) Consider a file system with a graphical user interface, such as Macintosh’s Finder, Microsoft’s Windows Explorer, or Linux’s KDE. The following objects were identified from a use case describing how to copy a file from a floppy disk to a hard disk: File, Icon, TrashCan, Folder, Disk, Pointer. Specify which are entity objects, which are boundary objects, and which are control objects.(3 Marks)
- b) The below figure is an example for one of the architectural styles. Based on the figure



- i. Identify the corresponding architectural style. (2 Marks)
 - ii. Draw the UML Component diagram for that architectural style. (2 Marks)
 - iii. Write down the features of that architectural style. (4 Marks)
 - iv. In which type of system this style is well suited. (1 Mark)
- c) Discuss how the MVC architecture helps or hurts the following design goals. (6 Marks)
- i. Extensibility (e.g., the addition of new types of views).
 - ii. Response time (e.g., the time between a user input and the time all views have been updated)
 - iii. Modifiability (e.g., the addition of new attributes in the model).
 - iv. Access control (i.e., the ability to ensure that only legitimate users can access specific parts of the model).

OR

- IV. a) Decomposing a system into subsystems reduces the complexity developers have to deal with by simplifying the parts and increasing their coherence. Decomposing a system into simpler parts usually results into increasing a different kind of complexity: Simpler parts also means a larger number of parts and interfaces. If coherence is the guiding principle driving developers to decompose a system into small parts, which competing principle drives them to keep the total number of parts small? Justify your answer. (2 Marks)
- b) If you are a developer and you are responsible to develop design patterns for the following problems. Suggest design pattern for each problem. (6 Marks)
- i. Shield the client from different platforms that provide different implementations for the same set of concepts.
 - ii. Convert the interface of a legacy class into a different interface expected by the client, so that the client and the legacy class can work together without changes.
 - iii. Decouple an interface from an implementation so that implementations can be substituted, possibly at runtime.
 - iv. Encapsulate requests so that they can be executed, undone, or queued independently of the request.
 - v. Represent a hierarchy of variable width and depth so that leaves and composites can be treated uniformly through a common interface.
 - vi. Reduce coupling between a set of related classes and the rest of the system.
- c) Compare between static and dynamic object modeling. (10 Marks)

- V. a) Describe the concept of mapping and mapping activities? (10 Marks)
- b) Explain configuration management? Discuss the management issues related to configuration management. (8 Marks)

OR

- VI. a) Describe integration testing strategies with examples. (10 Marks)
- b) Explain forward engineering and reverse engineering with examples? (8 Marks)