



UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 8 Examination in Engineering: November 2016

Module Number: EE8205

Module Name: Principles of Software Architecture

[Three Hours]

[Answer all questions, each question carries 10 marks]

Q1 For each question below, there is one correct answer. You get 1 mark for each correct answer. Please attach the answered paper (Page no. 1, Page no. 2 and Page no. 3) and write your index no.

01 Which of the following is correct regarding Enterprise Application Integration Styles?

- i). File Transfer and Shared Databases enable applications to share their data, but not their functionality.
- ii). Remote Procedure Invocation enables applications to share functionality, but tightly couples them in the process.
- iii). Both of the above.
- iv). None of the above.

02 Which of the following is the benefit of Web services being loosely coupled?

- i). The web service interface can change over time without compromising the client's ability to interact with the service.
- ii). Adopting a loosely coupled architecture tends to make software systems more manageable and allows simpler integration between different systems.
- iii). Both of the above.
- iv). None of the above.

03 Which of the following is correct regarding Transport Layer Security (TLS)?

- i). Secure end to end communication.
- ii). Not transparent through multiple transport protocols.
- iii). Can not specify different part of the message to secure in differently.
- iv). Relatively easy to attack.

04 Which of the following is true about Web Services RPC?

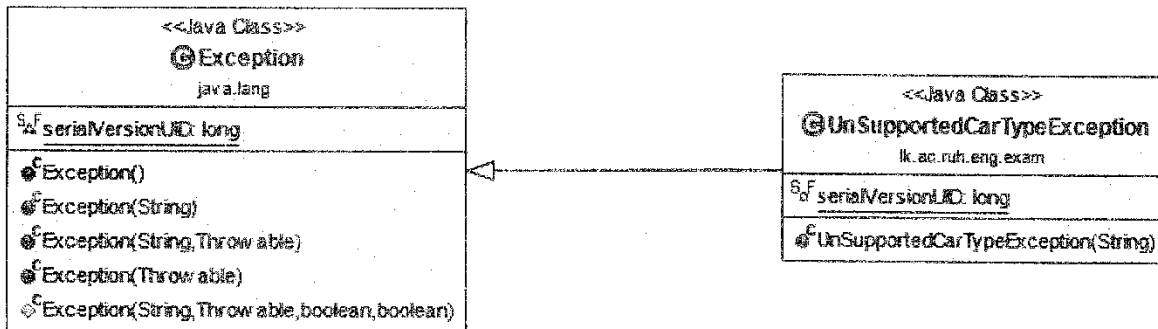
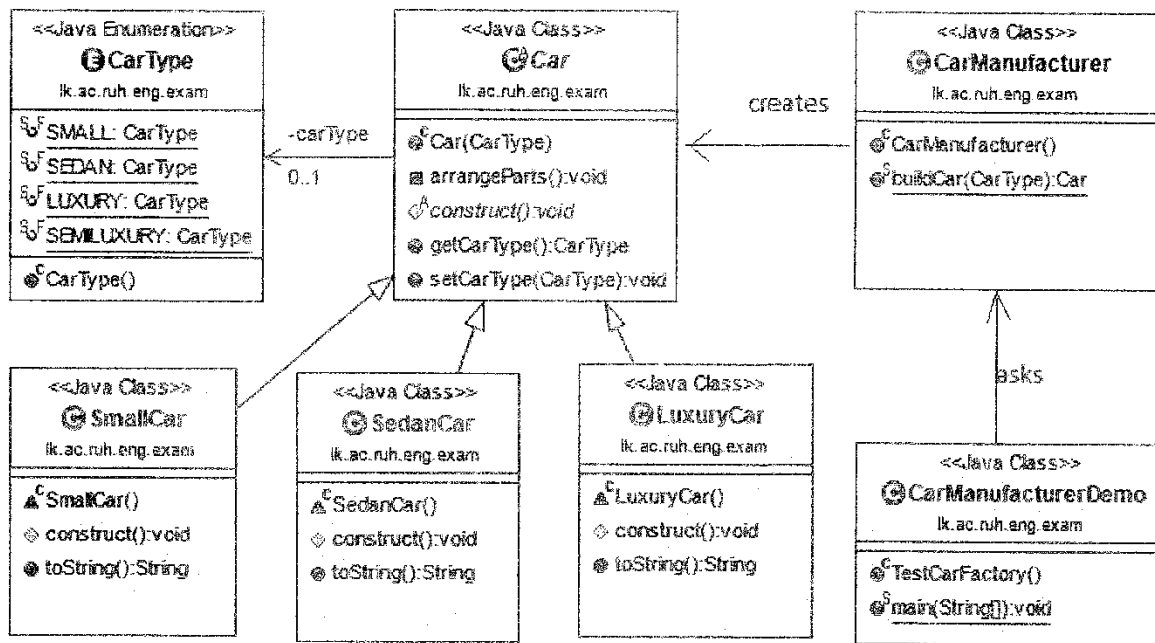
- i). Web services allow clients to invoke procedures, functions, and methods on remote objects using an XML-based protocol.

- ii). Remote procedures expose input and output parameters that a web service must support.
 - iii). A web service supports RPC by providing services of its own, equivalent to those of a traditional component, or by translating incoming invocations into an invocation of an EJB or a .NET component.
 - iv). All of the above.
- 05 Which of the following is true about behavioral characteristics of web services?
- i). Web Services uses XML at data representation and data transportation layers.
 - ii). A consumer of a web service is not tied to that web service directly.
 - iii). Businesses and the interfaces that they expose should be coarse-grained. Web services technology provides a natural way of defining coarse-grained services that access the right amount of business logic.
 - iv). All of the above.
- 06 Which of the following pattern is used when we need to decouple an abstraction from its implementation so that the two can vary independently?
- i). Bridge Pattern.
 - ii). Adapter Pattern.
 - iii). Prototype Pattern.
 - iv). Filter Pattern.
- 07 Which of the following describes the Observer pattern correctly?
- i). This pattern is used to get a way to access the elements of a collection object in sequential manner without any need to know its underlying representation.
 - ii). This pattern is used to reduce communication complexity between multiple objects or classes.
 - iii). This pattern is used to restore state of an object to a previous state.
 - iv). This pattern is used when there is one-to-many relationship between objects such as if one object is modified, its dependent objects are to be notified automatically.
- 08 Which of the following pattern is used to separate application's concerns?
- i). Visitor Pattern
 - ii). MVC Pattern
 - iii). Business Delegate Pattern
 - iv). Composite Entity Pattern

- 09 Integer class is an example of Decorator pattern.
- i). True
 - ii). False
- 10 Which of the following describes the Proxy pattern correctly?
- i). In this pattern a class represents functionality of another class.
 - ii). This pattern creates a chain of receiver objects for a request.
 - iii). This pattern provides a way to evaluate language grammar or expression.
 - iv). In this pattern a request is wrapped under an object as command and passed to invoker object.

- Q2 a) What do you mean by Design Patterns?
[1 Marks]
- b) The ABC (Pvt) Ltd is a car manufacturing company, which is able to manufacture 4 types of cars i.e. small, sedan, luxury and semi-luxury. But they have temporarily stopped manufacturing semi-luxury car type due to some manufacturing fault. Building a car requires several steps from allocating accessories to final makeup. These steps can be implemented in programming as methods and should be called while creating an instance of a specific car type. You have given a high level class diagram and code implementation for some classes [Refer code samples in figure Q2].
- i) What is the design pattern used in class diagram.
[0.5 Marks]
- ii) Complete code for ONE concrete implementation of class Car [You NO NEED to write code for all THREE]. The subclass constructor is responsible for setting correct car type and start executing its own car construct method. It should contain overridden version of method "toString()" to display that corresponding car construction is completed. [Hint : you may refer console output of demonstration class]
[2.5 Marks]
- iii) Implement the identified design pattern for the class CarManufacturer. [Hint: you may refer console output of demonstration class].
[2.5 Marks]
- c) Since existing manufacturing process is suitable for single location, the ABC (Pvt) Ltd is thinking to enhance the manufacturing process to support different car making styles for different country regions or locations. For example consider USA, ASIA and DEFAULT for all other countries.
- i) What is the design pattern you suggest for this enhancement?
[0.5 Marks]
- ii) Name the additional classes you suggest to add to the existing class diagram. What is the usage/responsibility of those classes. [You NO NEED to write complete code for these classes].
[1.5 Marks]
- d) List down the benefits of using design patterns?
[1.5 Marks]
- Q3 a) What do you mean by the term "Big Data"? Explain characteristic (V's) of big data.
[2.5 Marks]
- b) Describe the different data concept come with big data and provide an example for each.
[2 Marks]

- c) List down 5 data source of big data and well known 5 companies that use Hadoop?
[2 Marks]
- d) Briefly describe functionality of two core components in Hadoop framework.
[2 Marks]
- e) What is the difference between scale-up vs scale-out?
[1.5 Marks]
- Q4 a) What are the characteristic of enterprise application? Proved an example for enterprise and non-enterprise application.
[2 Marks]
- b) What is the ultimate goal of EAI (Enterprise Application Integration)?
[1 Marks]
- c) Briefly explain Enterprise Integration (EI) strategies.
[3 Marks]
- d) What are the Enterprise Application Integration (EAI) styles?
[2 Marks]
- e) What are the fundamental challenges in Enterprise Integration?
[2 Marks]
- Q5 a) Explain the term Channel, Message and Middleware.
[1.5 Marks]
- b) What is the difference between synchronous and asynchronous communication?
[1 Marks]
- c) What are the benefits and issues in Messaging System?
[4 Marks]
- d) A “tightly-coupled” integration, has following concerns. Explain how those concerns solve using enterprise integration.
- i) Platform Technology with internal representation of numbers and objects.
[0.5 Marks]
 - ii) The location with hard-coded machine addresses.
[0.5 Marks]
 - iii) All components have to be available at the same time.
[0.5 Marks]
 - iv) Data Format - A list of parameters and their types must match at both ends.
[0.5 Marks]
- e) List down the different message routing types.
[1.5 Marks]



```

Sample code:CarType

package lk.ac.ruh.eng.exam;

public enum CarType {
    SMALL, SEDAN, LUXURY, SEMILUXURY
}
  
```

```

Sample code:UnsupportedCarTypeException

package lk.ac.ruh.eng.exam;

public class UnsupportedCarTypeException extends Exception {

    private static final long serialVersionUID = 1L;

    public UnsupportedCarTypeException(String message) {
        super(message);
    }
}
  
```

Sample code:Car

```
package lk.ac.ruh.eng.exam;

public abstract class Car {

    private CarType carType = null;

    public Car(CarType carType) {
        this.carType = carType;
        arrangeParts();
    }

    private void arrangeParts() {
        System.out.println("Arrange parts for car type: " + carType);
    }

    /* Do specific car type construction in subclass */
    protected abstract void construct();

    public CarType getCarType() {
        return carType;
    }

    public void setCarType(CarType carType) {
        this.carType = carType;
    }

}
```

Sample code:CarManufacturerDemo with console output

```
package lk.ac.ruh.eng.exam;

public class CarManufacturerDemo {

    public static void main(String[] args) {
        try {
            System.out.println(CarManufacturer.buildCar(CarType.SMALL));
            System.out.println(CarManufacturer.buildCar(CarType.SEDAN));
            System.out.println(CarManufacturer.buildCar(CarType.LUXURY));

            System.out.println(CarManufacturer.buildCar(CarType.SEMILUXURY));
        } catch (UnsupportedCarTypeException e) {
            System.out.println(e.getMessage());
        }
    }

}
```

```
Arrange parts for car type: SMALL
Constructing a small car.
SMALL car is completed.
Arrange parts for car type: SEDAN
Constructing a sedan car.
SEDAN car is completed.
Arrange parts for car type: LUXURY
Constructing a luxury car.
LUXURY car is completed.
sorry, we have stopped manufacturing car type: SEMILUXURY
```

Figure Q2