

Overview of the Subject and Information for Exam Preparation

The following topics (and only them) will be covered in the exam: (1) JavaFX, (2) JDBC, (3) Socket Programming, (4) Web Services, and (5) AngularJS

1 Topic 1 – JavaFX

1.1 Overview of What We Have Covered

We have covered a lot of details about programming with JavaFX. Our coverage include the following:

- Chapter 1 – Basic JavaFX Program/Application
 - SampleFX is a basic JavaFX program
 - Elements of a basic program: stage (window), scene, and root
- Chapter 2 – Event Handling
 - Java 8 interfaces
 - Various ways to implement interfaces
 - Functional interfaces and lambda expressions
- Chapter 3 – Stage, Scene, layout Containers
 - Stage
 - Scene
 - Containers
 - * HBox
 - * VBox
 - * FlowPane,
 - * BorderPane
- Chapter 4 – Common Controls
 - Label
 - TextField
 - Button
 - TextArea
 - CheckBox
 - RadioBox, ToggleGroup
 - ChoiceBox
 - ComboBox

- Chapter 5 – TableView and Applications
 - Define table view
 - Define table columns
 - Set data items
 - Get the selected item
 - Set data items as a sorted filtered list
 - Binding comparator of sorted list to that of the table view
 - Listen to changes to a text field
 - Set customized sort order

The main contents of the JavaFX topic can be viewed as consisting of

1. The basics – which essentially include the SampleFX program, Label, TextField, Button, EventHandler<ActionEvent>, HBox, VBox, FlowPane and BorderPane
2. The use of TableView for data-intensive application development

These are the two parts that we will concentrate on for the exam.

1.2 Examinable Materials

1. **JavaFX Basics.** This part includes:

- SampleFX
- Scene, Stage
- HBox, VBox, FlowPane. **Note:** BorderPane is *not* examinable
- Label (with text label only)
- Button (with text label only)
- TextField
- TextArea
- EventHandler<ActionEvent>
- Exception handling in GUI applications
- Design and implementation of GUI applications using the above-mentioned controls

2. **TableView for Viewing and Manipulating Data Sources.** This part includes

- Define table view
- Define table columns
- Set the table view's data items
- Get selected item
- Add GUI controls to add, update and delete objects in a data source
- Relationships between the data sources, the data items of the table view (e.g. how to synchronize them)
- Design and implementation of GUI application using TableView
- **Note:** Customized sort is *not* examinable

1.3 Reference Materials for the Exam

While working with JavaFX programming, we often need to have access to the relevant API.

In the exam, a selected set of syntax is provided. For details of what are provided, see the Reference for Exam posted on LMS. The reference materials for JavaFX consists of two parts:

- Syntax for the basics of JavaFX programming
- Sample code to illustrate the syntax of commands to be used with TableView

1.4 Relevant Case Studies

- The BMI case study (Chapter 3)
- The Catalog case study (Chapter 5)

1.5 Some Sample Revision Questions:

1. Explain the general approach to application development with GUI using *JavaFX*.
Hint: Your answer should make use of the followings: Stage, Scene, layout containers, controls, and the event delegation mechanism.
2. Describe the following layout containers: FlowPane, HBox and VBox. Give examples to illustrate each of them.
3. What are the different ways to implement an event listener?
4. Explain functional interfaces and lambda expressions
5. Given an application with requirements comparable to that of the BMI case study. Design and implement that GUI application.
6. Explain the use of TableView for viewing and manipulating data sources.
7. Explain how to
 - Define a table view (including table columns and data items)
 - Add GUI controls to add, update and delete objects in the data source
8. Given an application with requirements comparable to that of the Catalog case study. Design and implement various parts of that GUI application.

2 Topic 2 - JDBC

2.1 Overview of What We Have Covered

- What is JDBC?
- Classes in JDBC API
- Basic SQL Statements
- SQL data types and their Java equivalents
- How to set up and execute SQL commands and queries
- How to process a query's result set
- How to use JDBC to define the data source controller to implement CRUD operations (Create Retrieve Update and Delete)

2.2 How to Prepare for the Exam

Strictly speaking, all the aspects listed in the Overview are relevant for the exam. The following table is meant to indicate what you can reasonably expect in the exam and how to prepare for it.

What JDBC is	You should be able to explain what JDBC for, what you can do with it
Classes in JDBC API	You should be able to identify some of the most commonly used classes of the JDBC API (that you have met in the CatalogDSC class for example)
Basic SQL Statements	You should be able to write simple insert, update, delete and select statements – similar to what we have in the CatalogDSC class.
SQL data types and their Java equivalents	You only need to know about the following basic types: integer, double, boolean, and strings.
How to set up and execute SQL commands and queries	You need to know how to make a connection (and to disconnect), create a Statement or a PreparedStatement object, and how to use them to execute commands and queries
How to process a query's result set	You need to know how to work with ResultSet class, but <u>not</u> the ResultSetMetaData.
How to use JDBC to define the data source controller	That is, you need to know how to implement CRUD operations, similar to what we have in the CatalogDSC class. Also, you need to think about the exception handling.

2.3 Relevant Case Study

The CatalogDSC class (version provided for chapter 8)

The CatalogDSC class provided for chapter 14 (“CatalogMPA-New Sample Code (with informative error messages”)

2.4 Reference Materials for the Exam

A sample program and the basic syntax for SQL statements will be provided with the exam paper. For details, see the Exam Reference Materials posted on LMS.

2.5 Sample Revision Questions

1. What is JDBC?
2. What are some of the classes in JDBC API?
3. Distinguish between a local server and a remote server.
4. Explain the main steps that we use to interact with a database.
5. How to connect to a database?
6. How to disconnect from a database?
7. Give examples to illustrate how to execute SQL statements (commands and queries) in a Java program using the Statement object.
8. Give examples to illustrate how to execute SQL statements (commands and queries) in a Java program using the PreparedStatement object.
9. Consider the Catalog case study. Assume that the connection to the database has been made. Write method to
 - (a) Add a product
 - (b) Update the price of a product
 - (c) Delete a product
 - (d) Find a product
 - (e) Retrieve all the product as an ArrayList of Products

For update and delete operations, what should we do if the product does not exist? Discuss this issue.

3 Topic 3 – Socket Programming

3.1 What We Covered

- What is socket programming?
- The basic steps in creating a server program, a client program and how they communicate with each other (this is illustrated by the DoubleServer1 and DoubleClient1)
- How to work with (create and start) threads (using Thread class or the Runnable interface)
- How to create a server program that can simultaneously serve multiple clients, with multiple services for each client
- How to terminate server programs and client program gracefully

3.2 How to Prepare for the Exam

What is socket programming about?	You should be able to explain what socket programming is for, what we can do with it
The basics of socket programming	<p>You should be able to explain the basics of socket programming.</p> <p>How we create a ServerSocket; how the server program waits for and accepts a client request; how the server communicate with the client.</p> <p>How a client requests a connection to a server program; how it communicates with the server.</p> <p>Note that all these features are covered in the DoubleServer1-DoubleClient1 example</p>
How to work with threads	How to define a thread class (e.g. a class that extends class Thread). How to use such a class to create and start a thread.
How to create a server program that can simultaneously serve multiple clients, with multiple services for each client	You should understand the techniques shown in the DoubleServer4-DoubleClient4 example, and how to use them to write similar programs

3.3 Useful Case Studies

- DoubleServer1-DoubleClient1
- DoubleServer4-DoubleClient4

3.4 Reference Materials for the Exam

See the Exam Reference Materials posted on LMS.

3.5 Sample Revision Questions

1. Explain the difference between socket programming and the “ordinary” programming that we have done in a first or second course in Java such as OOF or IOO.
2. Explain the basic techniques of socket programming, and illustrate them with a simple example.
3. *The Addition Server.* Write a server and a client program that behave as described below when viewed from the client side:
 - (a) The client first asks for connection to the server program
 - (b) The client asks the user for a number n_1 and send it to the server
 - (c) The client asks the user for a number n_2 and send it to the server
 - (d) Then reads the sum from the answer that the server sends back, and displays it on the screen

This request can be repeated as many times as the user wants to.

Assume that you can allow the client and server programs terminate ungracefully.

4. Modify the server program in the previous question so that it can serve multiple clients at the same time.

Note: For the exam, we will *not* be concerned with the issue of how to terminate server programs and client programs *gracefully*.

4 Topic 4 – Web Services

4.1 What We Have Covered

- What are web services? What are they for?
- SOAP-based web services
- REST and RESTfull web services
- Design of web services
 - HTTP methods and CRUD operations
 - Anatomies of REST requests and responses
- How JSON is used for data exchange
- How to implement and deploy web services (server side)
- How write Java Client programs to access web services

4.2 Preparation for the Exam

All the features listed above are examinable.

A good way to prepare for the exam is to study Chapter 13; in particular the Catalog case study.

4.3 Reference Materials for the Exam

See the Exam Reference Materials posted on LMS.

4.4 Some Sample Questions

1. Give a definition of web service.
2. Explain the fundamental insights of REST. Include concepts such as architectural style and constraints.
3. What are RESTful web services?
4. Explain the relationships between HTTP methods and CRUD operations. Illustrate how we can use these relationships to design the anatomies of REST requests and responses.
5. What is JSON? Explain how JSON can be used in web services.
6. Consider an example similar to the Catalog case study. Design and implement a web service to add an object. And then write a client program to add a product. Repeat for the other CRUD operations.
(You do not need to implement the three helper methods that are used to write Java web service client programs, e.g. `makeRESTRequest`.)
7. Consider the Catalog case study. Suppose we want a web service for the clients to do keyword search on product names, i.e. to return products whose name contain a specified string. How would you design and implement this web service?

5 Topic 5 – AngularJS

(Web Application Development with AngularJS (and Web Services))

5.1 What We Have Covered

- An example showing how to display an object
- An example how to display an array of objects
- The MVC Framework of AngularJS
 - What constitutes the view, the controller and the model
 - Basic structure and constructs of an AngularJS application
 - * HTML pages
 - * JavaScript controller
 - * Directives, e.g. ng-app, ng-model, ng-repeat, ng-click, ng-submit
 - * Expressions
- How to make HTTP calls to access web services from a JavaScript controller
- Building multi-page applications
- Building single-page applications

5.2 Preparation for the Exam

All the features listed above are examinable. One way to prepare for the exam is to study Chapters 15 and 16; in particular the Catalog case study.

5.3 Reference Materials for the Exam

See the Exam Reference Materials posted on LMS.

Note: HTML elements relevant to the exam are:

- Lists
- Tables
- Forms
- Input (type = “text”)
- Buttons (type= “submit”, and type = “button”)

5.4 Some Sample Review Questions

1. Give a simple example of AngularJS application and, using the example, explain the MCV framework of AngularJS
2. Give examples to show how to access web services to
 - Get an object
 - Get a array (list) of objects
 - Add an object
 - Update an object
 - Delete an object
3. Explain the design of an MPA
4. Discuss the advantages and disadvantages of an MPA
5. Implement various parts of an MPA
6. In assignment 2, for adding, editing and deleting operations, we display a message about whether the operation is successful or not. Why do we do this? And how?
7. Explain the purpose and the design of a typical SPA
8. Explain the AngularJS features needed to implement a SPA (besides those needed for MPA)
9. Explain the advantages of SPA

Note: In the exam, you will *not* be asked to write detailed code for the implementation of SPA.

