

Practical exam paper: Grads 2020 programme

Sep 2020
HSBC Technology Academy

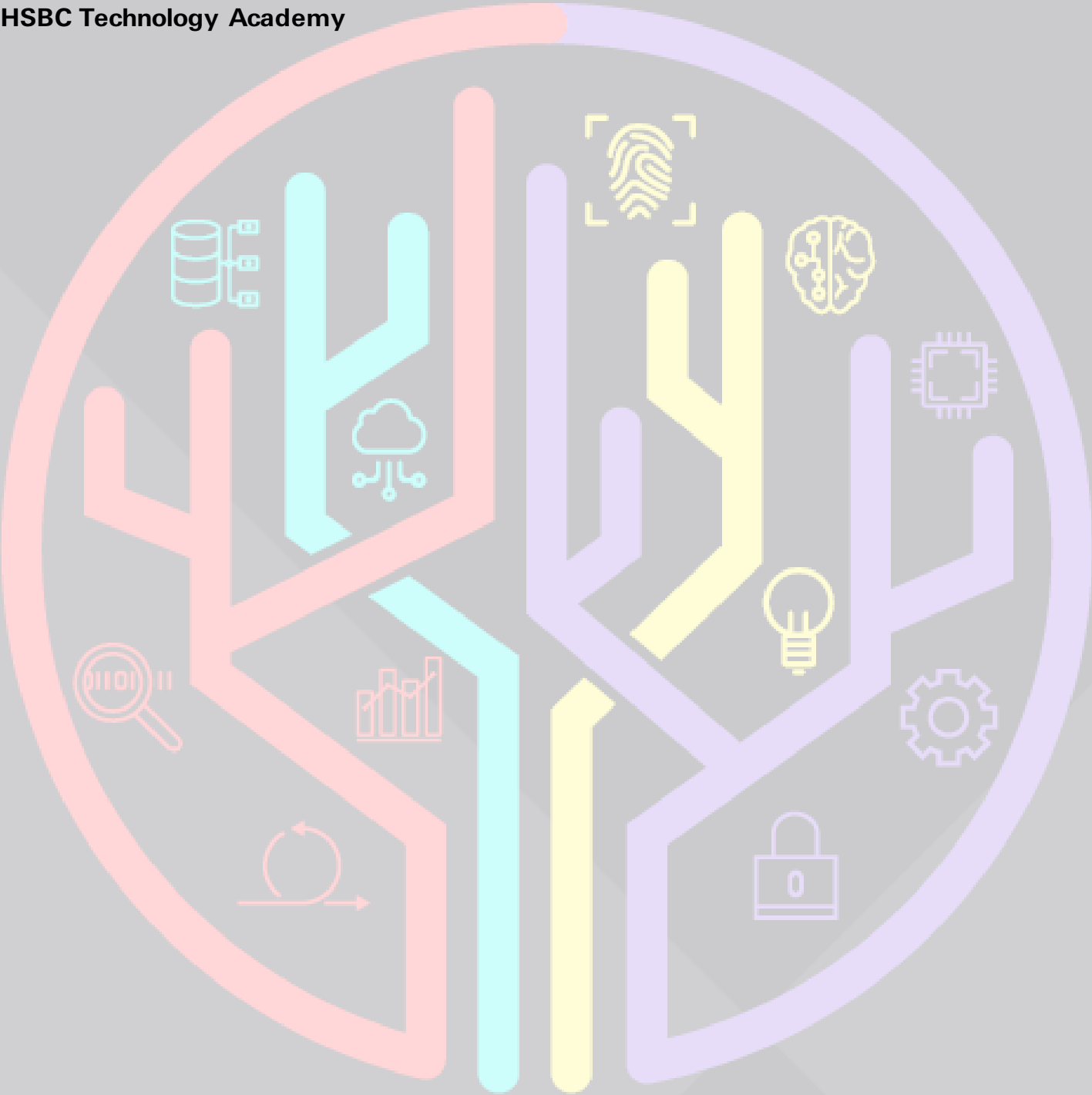


Table of contents

Notes	2
Practical exam paper: Grads 2020 programme	3
Problem description	3
Guidelines to follow	5



Notes

1. This is an open book exam
2. Duration: 90 minutes Max marks: 50
3. Please follow all coding standards and coding principles and good practices taught to you till date of exam
4. The solution should conform to 3-layer architecture (However, do not implement view layer for this exam)
5. Upload your solution to GITHUB once exam is done in your respective fork.
6. In each Java source file that you create, you should include below header in the code:
 - a. Author: <full name>
 - b. Purpose: <brief description of what program does>
7. Once all your sources are coded, tested and are done, upload Java and other **source** files to GITHUB "Practical_exam_<firstname>_<lastname>_<batchName>" folder.
(Do not upload JARs and other binaries which are part of your project).

Practical exam paper: Grads 2020 programme

Problem description

EveryDay GoodProducts Pvt. Ltd. is a retail company that transacts in different kinds of products across different categories. They have approached you to build an application (using Java) that generates a daily sales report. The detailed requirements are as follows:

◆ Data Storage

- There are 3 categories of products that the retailer is interested. These categories and information stored about products in the category are as follows:
 - **Food Items:** Each product in the food item category has - item code, item name, unit price, Date of manufacture, Date of expiry and Vegetarian (a Yes/No field), Quantity
 - **Apparel:** Each product in the apparel category has - item code, item name, unit price, size and material (either cotton or woolen), Quantity
 - **Electronics:** Each product in the electronics category has - item code, item name, unit price and warranty (in months), Quantity
- Example of how items related to electronics category are stored is as follows. Similar structure applies for food and apparel as well.

Item Code	Item Name	Unit Price	Warranty	Quantity
1013	TV	3000	12	100
1026	Mobile	8000	24	97
1114	Watch	1500	18	54

- Store this data using the data structures provided by the Collections framework

◆ Program output

- The program should accept the category as input (among food items, apparel and electronics) from the user and return a report of the top 3 items sold based on the quantity. Sample reports for each category are as shown below:

Food Items

Item Code	Item Name	Quantity Sold	Vegetarian
101	Milk	40	Yes

102	Curd	20	Yes
111	Cake	12	No

Apparel

Item Code	Item Name	Quantity Sold	Size	Material
222	T-shirt	50	Large	Cotton
432	Shirt	32	Medium	Cotton
325	Sweater	20	Medium	Woolen

Electronics

Item Code	Item Name	Quantity Sold	Warranty
1013	TV	100	12
1026	Mobile	97	24
1114	Watch	54	18

◆ Project Structure

- Design the Java project to follow a layered architecture. Keep in mind the following components:
 - Data Access layer – Should contain methods to add new items, and retrieve the data necessary for the report
 - Business logic layer – Should contain methods to invoke the data access layer and perform relevant business logic in order to add new items and also to retrieve data and generate the report
 - View layer – View layer is implemented as the Main class of the project. This layer accepts input from the user using the console and displays the report back on the console.
 - Entity classes – Appropriate entity classes need to be implemented
 - Exceptions – Appropriate exceptions must be raised. For example when trying to add a item if its itemcode already exists or when trying to retrieve a report of a category that doesn't exist.

◆ Deliverables

- The entire Java project comprising entities, exceptions, business logic layer, data access layer, etc.
- A README file that contains the following information

Layer	File	Description
Data Access Layer	XYZ.java	This file contains methods that do ...

Guidelines to follow

1. Follow appropriate naming and coding conventions in Java
2. Include appropriate comments at class and method level
3. Use layered architecture and ensure that each layer includes the right functionality
4. Ensure proper exception funneling/propagation across different layers
5. Follow best practices in terms of design patterns when creating the data access and business logic layers