**EVALUATION FORM (Object-Oriented programming with Java)**

Evaluator name and student number: Your name and student number here

Evaluation type: Self-evaluation / Peer-evaluation

Project’s repository name in Github: Name of the project repository at Github

Total evaluated points (max 50 points): Sum of points from the table below (Evaluation column)

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation targets** | **Points** | **Evaluation** | **Notes and justifications from the evaluator** |
| **Application description and code hosting at Github** | **5** | **5** |  |
| Readme file in Github with a description of the application (See guidance notes below) | 1 | 1 |  |
| Github Wiki pages are used to provide a description of the project/application and important information about it   * Purpose, goals, users (1) * Screen-shots of the UI showing key functionality (1) * Installation instructions (1) * Usage instructions (1) | 1-4 | 4 |  |
| **GUI implementation** | **9** | **9** |  |
| Control(s) (UI) to view data | 2 | 2 |  |
| UI to update data | 2 | 2 | When clicked on btnUpdateDVD |
| UI to delete data | 2 | 2 | When clicked on btnDeleteDVD |
| UI to add data | 3 | 3 | When clicked on btnAddDVD |
| **Database implementation** | **9** |  |  |
| SQL script that creates the database/tables is published at Github | 1 | 1 |  |
| Exception handling (try/catch blocks) is performed for Database operations | 2 | 2 | getAllDvd(), addDVD(), updateDVD(), deleteDVD() |
| JDBC calls used for CRUD operations  Creating/Adding data: 2 points  Reading data: 2 points  Updating data: 1 points  Deleting data: 1 points | 1-6 | 6 | addDvd (adding data to db)  deleteDvd (delete a movie from the db)  getAllDvd(reading data from db)  updateDvd(Updating data from db) |
| **Java OO implementation** | **18** |  |  |
| Amount of implemented Classes  1 Class: 1 points  2 Classes: 4 points  3 or more Classes: 6 points | 1,4,6 | 6 | Class names: 1. Dvd, 2. DvdQueries, 3. DvdApp |
| A Constructor with input parameters is used | 1 | 1 | Constructor name: Dvd |
| Access validators (private/public) properly used at a Class (attributes, methods) ensuring encapsulation and information hiding | 2 | 2 | Class name: Dvd, DvdApp |
| Setters, and Getters are used to ensure encapsulation and information hiding in all Classes | 1 | 1 | Method names: getName(), getDirector(), getGenre(), getYear(), getMovieid() |
| Setters methods contain data validation logic to ensure the program logic will work as expected | 1 | 0 | Method names |
| Exception handling (try/catch blocks) is used (ex: arithmetic, input mismatch).  2 points per type of exception (max two exception types) | 2,4 | 4 | addDvd, deleteDvd, updateDvd, getAllDvd |
| The following keywords are properly used in a Class:   * this (1 point) * static (1 point) * final (1 point) | 1-3 | 3 | This: Dvd  Static: DvdApp  Final: DvdApp |
| **Coding conventions – good practices** | **5** |  |  |
| Naming: variables, classes and methods are named consistently, using proper cases and suitable names | 1 | 1 | DvdApp, Dvd, DvdQueries,  addDvd, updateDvd, deleteDvd, getAllDvd |
| Indent style: code is indented properly | 1 | 1 |  |
| The code contains comments allowing other developers to understand the logic/thoughts under Classes and methods  Some comments can be found in the code: 1 point  All Classes and methods are commented with useful info: 3 points | 1,3 | **3** |  |
| **Delivery and evaluation** | **4** | 4 |  |
| Evaluation of other student’s project is delivered with **valid/correct** evaluations | 4 | 4 | Name of students being evaluated |
| Own project has **not** been delivered on-time (according to the deadline set by the teacher) | (-) 5 |  | Deduct five points from your self-evaluation |

**Guidance notes:**

|  |  |
| --- | --- |
| **Database implementation** |  |
| SQL script that creates the database/tables | .sql file containing SQL sentences that can be imported into your database server. Primary and foreign keys (if applicable) need to be defined in the script |
| **Application description and code hosting at Github** |  |
| Readme file | Readme file having similar content as explained at:  <https://guides.github.com/features/wikis/> |
| Screenshots | This is important as someone (potential employer?) may not have the time to install the app and see how it works. Publishing the screenshots of your app will give a good idea of how the application works |
| Installation instructions | Make sure the instructions are clear and complete, in such a way that anyone wanting to test your app will be able to setup the environment and start the app |