

code.sprint^{MT}

TASKS BOOKLET

- Final Round -

Post-Secondary Category

2019



DIRECTORATE FOR LEARNING &
ASSESSMENT PROGRAMMES

Final Round Schedule

Time Allotted	Analysis	Task	Submit	Break
15:00 – 15:05	Task			
15:05 – 16:35		Task		
15 minutes				Interval
16:50 – 18:20		Task		
18:20 – 18:25			Task	
FINALISTS ARE ANNOUNCED				

JACK: Musical Instruments Renting App (180 min)

JACK, a musical instrument rental intermediary, situated in the centre of Malta, has opened its doors last week. The business owners Josmar, Anthony, Claire and Keith (JACK) need a simple application to manage the rentals of their musical instruments which are supplied by their suppliers to their clients.



By using CLI screens or GUI components, develop an application to store records about clients, musical instruments and rentals according to the rules below:

Program Rules:

- The application should support the **THREE** main entities having the corresponding fields as follows:
 - **Client** (id, name, surname, email, contactNum, preferredBrand, state)
 - **Instrument** (id, name, make, family, costPerDay, supplierName, state, borrowedTimes)
 - **Rental** (rentalNum, rentalDate, returnDate, clientId, instrumentId, totalCost)
- The application should be menu driven and should include the following menus: **Main Menu, Clients Menu, Instruments Menu, and Rentals Menu**. *Check table 1 for the menu details.*
- The application should include the validation rules as listed in *table 2* and proper messages should be displayed accordingly.
- Clients can rent more than one different instrument at the same time.
- An instrument cannot be rented if still rented to a client; i.e. instrument state is "unavailable"
- You cannot delete instrument records if they are on loan.
- You cannot delete client records if they have instrument on loan; i.e. client state is "pending"
- You cannot delete rental records if the rental is overdue and the instrument is not yet returned; i.e. return date is empty
- The records, as shown in tables 3, 4, & 5 below, should be inputted prior submitting your work.
- Records should be written and read from text or object files.

<pre>-- Main Menu -- Clients Instruments Rentals Exit</pre>	<pre>-- Clients Menu -- Add clients Edit clients Delete clients List all clients Return to main menu</pre>
<pre>-- Instruments Menu -- Add instruments Edit instruments Delete instruments List all instruments List most borrowed instruments Return to main menu</pre>	<pre>-- Rentals Menu -- Add rentals Delete rentals List all rentals List pending rentals Return to main menu</pre>

Table 1: Menu Options

Validation Rules	
Menus	The menu items chosen should be a valid option
Client Records	<p><i>id</i> should be unique and cannot contain duplicates</p> <p><i>name</i> & <i>surname</i> cannot contain non-digit characters</p> <p><i>contactNum</i> contains only digits</p> <p><i>state</i> should be 'Settled' or 'Pending' according to whether the client has a pending rentals or not.</p>
Instrument Records	<p><i>id</i> should be incremented automatically and cannot contain duplicates</p> <p><i>name</i>, <i>make</i> & <i>family</i> cannot contain digit characters</p> <p><i>costPerDay</i> should contain only numbers with a decimal point</p> <p><i>supplierName</i> should contain a supplier name.</p> <p><i>state</i> should be 'Available' or 'Unavailable' only, according to whether it is rented to a client or not.</p> <p><i>borrowedTimes</i> should be incremented automatically by one every time the instrument is chosen for rent</p>
Rental Records	<p><i>id</i> should be incremented automatically and cannot contain duplicates</p> <p><i>rentalDate</i> should be in the form of dd/mm/yyyy</p> <p><i>returnDate</i> should be in the form of dd/mm/yyyy and cannot be less than the rental date</p> <p><i>clientId</i> should be an existent Client ID and have an Active account</p> <p><i>instrumentId</i> should be an existent Instrument ID and is available</p> <p><i>totalCost</i> should be calculated automatically; <i>i.e.</i> $rental\ days * costPerDay$</p>

Table 2: Validation Rules

Client						
id	name	surname	email	contactNum	preferredBrand	State
123499M	Peter	Cauchi	pcauchi@gmail.com	79797979	Yamaha	Settled
451298M	Alister	Borg	bali@gmail.com	79891254	Onkyo	Pending
865384M	Alexia	Tabone	tabonealexia@gmail.com	99531736	Roland	Settled

Table 3: Client Records

Instrument							
id	name	make	family	costPerDay	supplierName	state	borrowed Times
01	Piano	Steinway	Strings	50.00	Ministry of Music	Unavailable	5
02	Electric Guitar	Roland	Strings	40.00	MI rentals	Available	3
03	Bass Drum Kit	Tama Drums	Percussion	45.00	Instrument Supplies	Available	12

Table 4: Instrument Records

Rental						
rentalNum	rentalDate	returnDate	clientId	instrumentId	totalCost	
01	06/04/2019	09/04/2019	865384M	01	100.00	
02	08/04/2019	11/04/2019	451298M	03	135.00	
03	10/04/2019		451298M	01		

Table 5: Rental Records

Name the class containing the main method **RunApp**
 Submit your work in a folder called **JACK_INDEXNUM**
 e.g. JACK_0025 or JACK_0004

Assessment Rubric

Program Functionality	User Friendly GUI Interface (use of basic AWT, Swing or JavaFX GUI)	Code Efficiency	Proper use of In-line Text (Comments)	Use of Proper Conventions (Camel Case, meaningful variable names etc.)
Name of Folder & Class/es	User Input	Suitable Prompts / Messages displayed	Validation (Menu Options)	Validation (Clients Records)
Validation (Instruments Records)	Validation (Rentals Records)	Use of a proper sorting algorithm	Use of proper Data Structures	Use of Files (Writing/Reading)
Arithmetic Process (Autoincrement of certain fields)	Arithmetic Process (total rental cost)	Use of separate classes (to manage entity records in a collection)	Preloaded Records	Other Features (not listed in the task)

Maximum Score: 38
+ 2 for every extra feature

0 – Not Satisfactorily | 1- Partly Satisfactorily | 2- Entirely Satisfactorily

