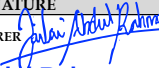

 <p>POLITEKNIK BRUNEI</p>	<b>SCHOOL OF POLITEKNIK BRUNEI</b> <b>WEEKLY MODULE PLAN</b>	REFERENCE NUMBER: PB/AS/LP/002 VERSION: 1.2 REVISION DATE: 06/10/2018 EFFECTIVE DATE: 1/1/2019													
MODULE CODE/NAME	NS4307, Network Programming	TOPIC	Java Persistence API												
SEMESTER	Semester 2, 2021 / 2022	GROUP CODE/INTAKE	DITN12 (Intake 11), DITN10R (Intake 9)												
SCHOOL/DEPT	School of ICT	WEEK	Week 12 - 10/04 - 16/04												
<b>PEDAGOGICAL APPROACH</b>															
<input type="checkbox"/> Knowledge Building <input type="checkbox"/> Discussion <input checked="" type="checkbox"/> Presentation <input checked="" type="checkbox"/> Blended Learning <input type="checkbox"/> Co-operative/ Collaborative Learning <input type="checkbox"/> Debate <input type="checkbox"/> Experimental Learning <input checked="" type="checkbox"/> Case Studies <input type="checkbox"/> Others, please specify _____															
<b>RESOURCES</b>															
Learning management system															
<table border="1" style="width: 100%;"> <thead> <tr> <th data-bbox="172 573 456 595">LESSON LEARNING OBJECTIVES</th> <th data-bbox="456 573 962 595">METHOD OF INSTRUCTION/LEARNING ACTIVITIES</th> <th data-bbox="962 573 1319 595">REFLECTIVE EVALUATION</th> </tr> </thead> <tbody> <tr> <td data-bbox="172 595 456 931">           At the end of this lesson, the student should be able to:            1) Apply Java Persistence API         </td> <td data-bbox="456 595 962 931"> <b>Session 1:</b>            Practical (1 hour 45 minutes):            - Show how to make POJO to structure data in database.            - Show what @Entity and @Table annotation does.            - Show what @Id, @GeneratedValue and @Column annotation does.            - Emphasis that it is required to implement a no-arg constructor in POJO.            - Show how to create table relationships between two POJO.             Exercise (15 minutes)            - Complete the implementation of Student based on the requirements:            1) id is the primary key            2) name should not be null         </td> <td data-bbox="962 595 1319 931">           Some students were having error when implementing this. Due to following reasons:            1) Created the model class in a different package from the actual application implementation.            2) Forgot to type in the @Entity and @Table annotation.            3) Not connected to the internet, even though it has been always been emphasised that the mysql server database. jailanirahman.com is hosted in Singapore.            4) Did not install the Spring jpa and mysql connector due to absent from previous session.            5) Common basic Java mistakes.             Reasons #3 and #5 are very concerning, since they are in semester 4 already where they still don't have good grasp of the basics that they have learned previously.         </td> </tr> <tr> <td data-bbox="172 931 456 1234">           At the end of this lesson, the student should be able to:            1) Apply Java Persistence API         </td> <td data-bbox="456 931 962 1234"> <b>Session 2:</b>            Practical (1 hour 30 minutes):            - The following are implementations for Student            - Show how to implement the interface in to handle POJO            - Show how to use DAO in controller            - Show how to retrieve all entities using DAO            - Show how to retrieve one entity using DAO            - Show how to implement our own method in DAO            - Show how to add data to database            - Show how remove data from database            - Show how to update data from database             Exercise (30 minutes):            - Complete the CRUD implementation for Group         </td> <td data-bbox="962 931 1319 1234">           Realised that the previous exercise will caused an issue during this session. Have to spent 30 minutes rectifying the code to prevent the issue to arise. But due to this, did not managed to cover deleting and updating data. Will cover this on next week session.         </td> </tr> <tr> <td data-bbox="172 1234 456 1357">           At the end of this lesson, the student should be able to:         </td> <td data-bbox="456 1234 962 1357"> <b>Session 3:</b> </td> <td data-bbox="962 1234 1319 1357"></td> </tr> </tbody> </table>				LESSON LEARNING OBJECTIVES	METHOD OF INSTRUCTION/LEARNING ACTIVITIES	REFLECTIVE EVALUATION	At the end of this lesson, the student should be able to: 1) Apply Java Persistence API	<b>Session 1:</b> Practical (1 hour 45 minutes): - Show how to make POJO to structure data in database. - Show what @Entity and @Table annotation does. - Show what @Id, @GeneratedValue and @Column annotation does. - Emphasis that it is required to implement a no-arg constructor in POJO. - Show how to create table relationships between two POJO.  Exercise (15 minutes) - Complete the implementation of Student based on the requirements: 1) id is the primary key 2) name should not be null	Some students were having error when implementing this. Due to following reasons: 1) Created the model class in a different package from the actual application implementation. 2) Forgot to type in the @Entity and @Table annotation. 3) Not connected to the internet, even though it has been always been emphasised that the mysql server database. jailanirahman.com is hosted in Singapore. 4) Did not install the Spring jpa and mysql connector due to absent from previous session. 5) Common basic Java mistakes.  Reasons #3 and #5 are very concerning, since they are in semester 4 already where they still don't have good grasp of the basics that they have learned previously.	At the end of this lesson, the student should be able to: 1) Apply Java Persistence API	<b>Session 2:</b> Practical (1 hour 30 minutes): - The following are implementations for Student - Show how to implement the interface in to handle POJO - Show how to use DAO in controller - Show how to retrieve all entities using DAO - Show how to retrieve one entity using DAO - Show how to implement our own method in DAO - Show how to add data to database - Show how remove data from database - Show how to update data from database  Exercise (30 minutes): - Complete the CRUD implementation for Group	Realised that the previous exercise will caused an issue during this session. Have to spent 30 minutes rectifying the code to prevent the issue to arise. But due to this, did not managed to cover deleting and updating data. Will cover this on next week session.	At the end of this lesson, the student should be able to:	<b>Session 3:</b>	
LESSON LEARNING OBJECTIVES	METHOD OF INSTRUCTION/LEARNING ACTIVITIES	REFLECTIVE EVALUATION													
At the end of this lesson, the student should be able to: 1) Apply Java Persistence API	<b>Session 1:</b> Practical (1 hour 45 minutes): - Show how to make POJO to structure data in database. - Show what @Entity and @Table annotation does. - Show what @Id, @GeneratedValue and @Column annotation does. - Emphasis that it is required to implement a no-arg constructor in POJO. - Show how to create table relationships between two POJO.  Exercise (15 minutes) - Complete the implementation of Student based on the requirements: 1) id is the primary key 2) name should not be null	Some students were having error when implementing this. Due to following reasons: 1) Created the model class in a different package from the actual application implementation. 2) Forgot to type in the @Entity and @Table annotation. 3) Not connected to the internet, even though it has been always been emphasised that the mysql server database. jailanirahman.com is hosted in Singapore. 4) Did not install the Spring jpa and mysql connector due to absent from previous session. 5) Common basic Java mistakes.  Reasons #3 and #5 are very concerning, since they are in semester 4 already where they still don't have good grasp of the basics that they have learned previously.													
At the end of this lesson, the student should be able to: 1) Apply Java Persistence API	<b>Session 2:</b> Practical (1 hour 30 minutes): - The following are implementations for Student - Show how to implement the interface in to handle POJO - Show how to use DAO in controller - Show how to retrieve all entities using DAO - Show how to retrieve one entity using DAO - Show how to implement our own method in DAO - Show how to add data to database - Show how remove data from database - Show how to update data from database  Exercise (30 minutes): - Complete the CRUD implementation for Group	Realised that the previous exercise will caused an issue during this session. Have to spent 30 minutes rectifying the code to prevent the issue to arise. But due to this, did not managed to cover deleting and updating data. Will cover this on next week session.													
At the end of this lesson, the student should be able to:	<b>Session 3:</b>														
<b>NAME &amp; SIGNATURE</b>															
MODULE LECTURER  <b>Jailani Abdul Rahman</b>		PROGRAMME LEADER / ASSISTANT HOS (aHOS) / HEAD OF SCHOOL (HOS)  <b>Jamiatul Zubriah</b>													
DATE: <b>13/04/2023</b>		DATE: <b>15/04/2023</b>													
<b>COMMENTS BY PROGRAMME LEADER/ ASSISTANT HOS (aHOS)/ HEAD OF SCHOOL (HOS) (If any)</b>															
Keep up your good work cg. You able to identify the issues and solve it well.															