**Familiarity Review**

**Name:** Nkenta Uchechukwu Ferdinand

**Date:** Oct 25th, 2019

**Week:** 6

**Coding Topic:** **Hibernate**

**Description of Understanding: Hibernate** ORM which the short form is just **Hibernate** is known to be an object-relational mapping tool for the **Java.** Its job is to provide a framework that maps an object-oriented domain model to a relational database. This is Licensed under an open source GNU Lesser General Public License (LGPL) which means it is always available for free downloads. The use of this hibernate generates SQL call and as well as a relieve to developers from manually handling and object conversion of the result set.

**Why do we need Hibernate in Java?**

* It provides a flexibility for changing the database when needed.
* It reduces the lines of code by maintaining object-table mapping itself and returns result to application in form of **Java** objects.
* It relieves programmer from manual handling of persistent data, hence reducing the development time and maintenance cost.

**Overall:** Hibernate is known for its stability and quality of results. More developers worldwide use and support it.

|  |  |
| --- | --- |
| **File** | **Git Link** |
| **Hibernate** | <https://github.com/nkenta/cit360/tree/master/Hibernate> |
| **CreateData.java** | [CreateData.java](https://github.com/nkenta/cit360/blob/master/Hibernate/CreateData.java) |
| **Employee.java** | [Employee.java](https://github.com/nkenta/cit360/blob/master/Hibernate/Employee.java) |
| **HibernateUtil.java** | [HibernateUtil.java](https://github.com/nkenta/cit360/blob/master/Hibernate/HibernateUtil.java) |
| **Hibernate.cfg.xml** | [hibernate.cfg.xml](https://github.com/nkenta/cit360/blob/master/Hibernate/hibernate.cfg.xml) |

|  |  |
| --- | --- |
| **What should I be looking for?** | **Sandbox or Your code?** |
| Here I have three Java class which each of this class performs a specific action. In the **CreateData.java:** Here you will see how I was able to add information that are to be displayed. If everything goes well, the system will print out a success message that my insert was successful.  **The Employee.java:** controls the set and get of the employee’s name, address, and number and are given a private ID.  **HibernateUtil.java:** This is where the configuration of the operation is made. The ServiceRegistry manages the service and at the same time act as inversion-of-control container.  **The SessionFactory:** This is the interface. It contains all the Database related property details which are pulled from **hibernate.cfg.xml** or **hibernate properties** file.  Also, it is important to remember that for the hibernate to work effectively a set of configurations settings related to the database and other related parameters are needed and that where the **hibernate.cfg.xml** comes in. Here I have the configurations needed to make sure that everything works according to how it was programmed. | Mine |