

DAY 2

Benefits of Hibernate

Benefits of Hibernate:

- Hibernate handles all low-level SQL queries
- It minimizes the JDBC code to be written
- It provides Object to Relational Mapping(ORM)

Let us understand what ORM means-

ORM stands for "**Object to Relational Mapping**" where

- The **Object** part is the one you use with your programming language.
- The **Relational** part is a Relational Database Management System i.e., database
- And finally, the **Mapping** part is where you do a bridge between your objects and your tables.

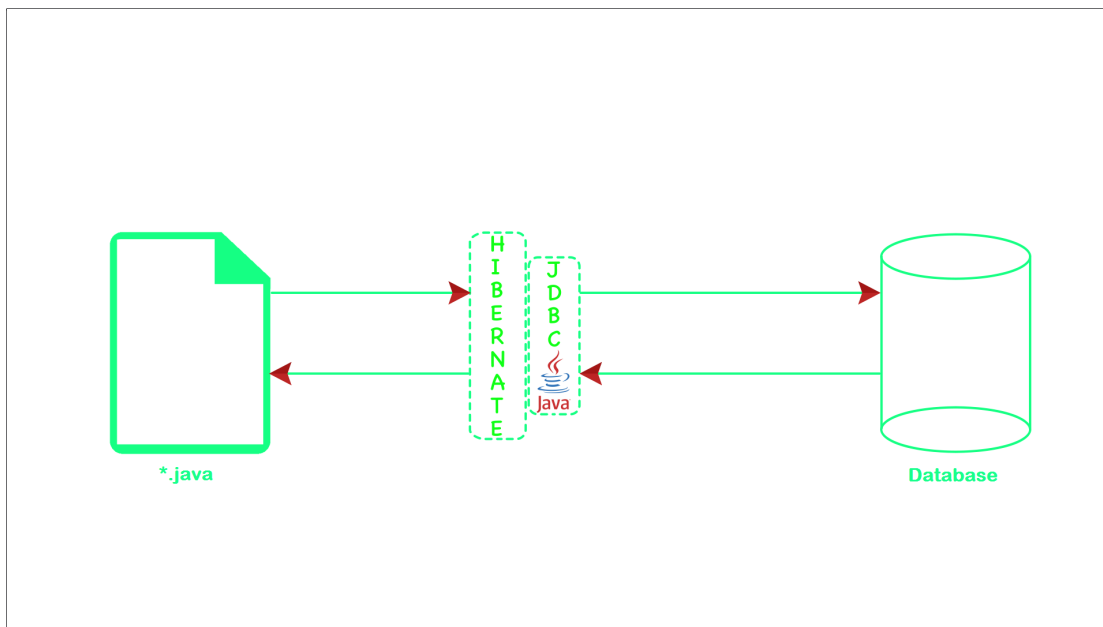
So in other words, Object-Relational Mapping (ORM) is a technique that lets you query and manipulates data from a database using an object-oriented paradigm.

And we can perform this mapping either through XML configuration or Annotations.

In JDBC sessions we had seen how we can share data between the database and the java applications.

Now with the help of Hibernate, we can make the job easier.

Hibernate internally makes use of **JDBC APIs** to establish connections with the database.



Important things to remember:

- Hibernate makes use of JDBC for all database connections
- Hibernate is another layer of extraction on top of JDBC
- When java application uses hibernate framework, the java application will store and retrieve objects using the Hibernate APIs
- Hibernate APIs uses JDBC APIs in the background for all operations to be done on the database

Now let us try to optimise the code which we had written in the previous session-

Student.java

```
package com.tapacad.application;

import java.io.Serializable;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import com.tapacad.model.Student;

public class Program1 {

    public static void main(String[] args) {
        SessionFactory sessionFactory = null;
        Session session = null;

        try {
            // Create Session Factory
            sessionFactory = new Configuration()
                            .configure()
                            .addAnnotatedClass(Student.class)
                            .buildSessionFactory();

            // Create Session
            session = sessionFactory.openSession();

            // Create Transaction
            Transaction transaction = session.beginTransaction();

            // CRUD Operations
            Student s1 = new Student(3, "charli", "charli@gmail.com");
            Serializable id = session.save(s1);
            System.out.println(id);

            transaction.commit();
        } finally {
            // Closing Resources
            session.close();
            sessionFactory.close();
        }
    }
}
```

Most of the code which we have written remains the same except for the CRUD operations so wouldn't it be better if we can create a shortcut to generate the code?

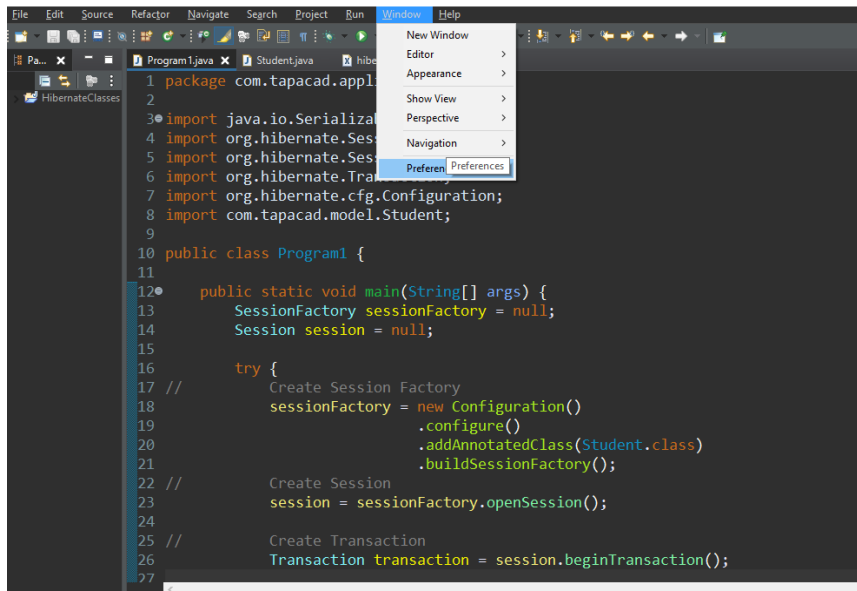
Let us try that now-

Step 1 :

Copy the code

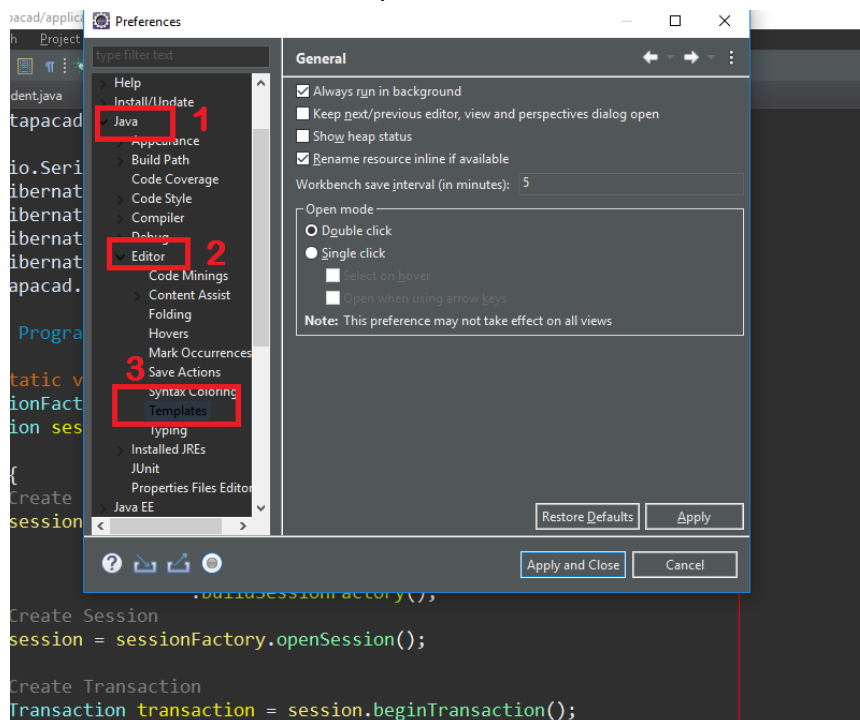
Step 2:

Click on the **Windows** tab and then click on **Preferences**



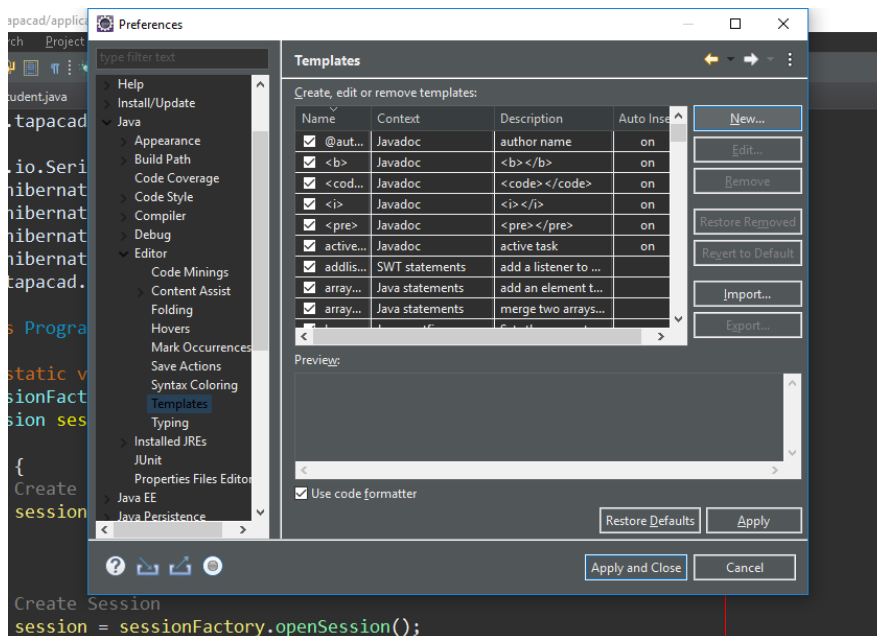
Step 3:

Click on the **Java** dropdown and click on **Editors** button and then click on **Templates**



Step 4:

Click on New, a pop-up window will be opened



Step 5:

Fill in the details as shown below and click on **Ok** and then click on **Apply and close**

