Hibernate.cfg.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<hibernate-configuration>
   <session-factory>

        <!-- Database Connection settings -->
       <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property
>
       <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernate</pro
perty>
       <property name="hibernate.connection.username">root</property>
       <property
name="hibernate.connection.password">Gopi@9201</property>

       <!-- JDBC Connection pool -->
       <property name="connection.pool_size">10</property>

       <!-- SQL dialect -->
       <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

       <!-- Logging -->
       <property name="show_sql">true</property>
```

```
        <!-- Automatically create/update table -->
        <property name="hbm2ddl.auto">update</property>

         <mapping class="com.example.model.Citizen"/>
        <mapping class="com.example.model.Passport"/>
    </session-factory>
</hibernate-configuration>
```

HibernateUtil:

```
package com.example.util;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class Hibernateutil {
        private static final SessionFactory sessionFactory= new
Configuration().configure().buildSessionFactory();
        public static SessionFactory getSessionFactory() {
                return sessionFactory;
        }
}
```

Citizen.java:

```
package com.example.model;

import jakarta.persistence.CascadeType;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.OneToOne;
@Entity
public class Citizen {
   @Id
   private int id;
   private String name;
   @OneToOne(cascade = CascadeType.ALL)
   @JoinColumn(name = "passport_id", referencedColumnName = "id")
   private Passport passport;
        public Citizen(int id, String name, Passport passport) {
                this.id = id;
                this.name = name;
                this.passport = passport;
        }
        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public Passport getPassport() {
                return passport;
        }
        public void setPassport(Passport passport) {
                this.passport = passport;
        }
}
```

Passport.java:

```
package com.example.model;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.OneToOne;
@Entity
public class Passport {
   @Id
   private int id;
   private String passportNumber;
   @OneToOne(mappedBy = "passport")
   private Citizen citizen;
        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
```

```
	public String getPassportNumber() {
		return passportNumber;
	}

	public void setPassportNumber(String passportNumber) {
		this.passportNumber = passportNumber;
	}

	public Citizen getCitizen() {
		return citizen;
	}

	public void setCitizen(Citizen citizen) {
		this.citizen = citizen;
	}
}
```

App.java:

```
package com.example;
import com.example.model.Citizen;
import com.example.model.Passport;
import com.example.util.Hibernateutil;
import org.hibernate.Session;
import org.hibernate.Transaction;
public class App {
	public static void main(String[] args) {
```

```
                     Session session = 
Hibernateutil.getSessionFactory().openSession();
                     Transaction tx=session.beginTransaction();

                 Passport passport = new Passport();
                 passport.setId(101);
                 passport.setPassportNumber("1234567890");

                 Citizen citizen = new Citizen(1,"Ravi Kumar", passport);

                 passport.setCitizen(citizen);

                 session.persist(citizen);

                 tx.commit();
                 session.close();

                 System.out.println("Bi-directional: Citizen and Passport saved.");
         }

}
```