Association Mappings:

The mapping of associations between entity classes and the relationships between tables is the soul of ORM. Following are the four ways in which the cardinality of the relationship between the objects can be expressed. An association mapping can be unidirectional as well as bidirectional.

**One to Many mapping:**

* One pojo class is related with multiple Pojo classes.
* In parent pojo class we need to take one collection property. In the mapping file of parent pojo class we need to configure that collection details. Collection can be Set, List, or Map.
* While mapping collection details, we need to add **cascade** attribute to transfer the operations on parent object to child objects.
* Default value for cascade is none, which means even though relationship is exist, the operations we are doing on parent object will not transfer to child object.
* In child class mapping file we need to take <key> tag to configure foreign key column.
* Consider 2 pojo classes, Student and Course. For both pojo classes we need to take a separate mapping file. Configure these 2 mapping file details in hibernate.cfg.xml file.

**One to Many mapping:**

Applying both one to many and many to one is nothing but one to many bidirectional.