Coupling

Coupling refers to the degree of direct knowledge that one element has of another. In other words, how often do changes in class A force related changes in class B.

They are two types of coupling

1.tight coupling

2. loose coupling

**Tight coupling :**

In general, Tight coupling means the two classes often change together. In other words, if A knows more than it should about the way in which B was implemented, then A and B are tightly coupled.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* first class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**class** Bus {

**public** **void** start() {

System.***out***.println("bus trip started..........");

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** TigCouple { // the classes TightCoupling and Bus are tightly coupled

**public** **static** **void** main(String[] args) {

Bus bus = **new** Bus();

}

}

**Loose coupling :**

loose coupling means they are mostly independent. If the only knowledge that class A has about class B, is what class B has exposed through its interface. then class A and class B are said to be loosely coupled.

\*\*\*\*\*\*\*\*\*\*\*\* interface \*\*\*\*\*\*\*\*\*

**interface** Transport {

**void** start();

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\* BUS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**class** Bus1 **implements** Transport {

@Override

**public** **void** start() {

System.***out***.println("journy started in BUS.............");

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TRAIN \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**class** Train1 **implements** Transport {

@Override

**public** **void** start() {

System.***out***.println("journy started in TRAIN......");

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CommonClass \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**class** MyTravel {

**private** Transport transsport;

**public** MyTravel(Transport transsport) {

**this**.transsport = transsport;

}

**public** **void** start() {

transsport.start();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MAIN CLASS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** **class** LooseCoupling {

**public** **static** **void** main(String[] args) {

Bus1 bus = **new** Bus1();

Train1 train = **new** Train1();

MyTravel my = **new** MyTravel(train);

my.start();

}

}