All good software design will go for **high cohesion** and **low coupling**.

**Cohesion** refers to what the class (or module) has to do. Low cohesion would mean that the class does a great variety of actions and is not focused on what it should do. High cohesion would then mean that the class is focused on what it should be doing, i.e. only methods relating to the intention of the class.

**Example Problem**

Class below is downloading from internet, parsing data and storing data to database. The responsibilities of this class are not really related. This is not cohesive class.

class DownloadAndStore{

void downloadFromInternet(){

}

void parseData(){

}

void storeIntoDatabase(){

}

void doEverything(){

downloadFromInternet();

parseData();

storeIntoDatabase();

}

}

##### Solution This is a better way of approaching the problem. Different classes have their own responsibilities.

interface Downloader {

public void download() {

}

}

interface Parser {

public void parse() {

}

}

interface Storer {

public void store () {

}

}

class InternetDownloader implements Downloader {

public void download () {

}

}

class XMLParser implements Parser {

public void parse () {

}

}

class DatabaseStorer implements Storer {

public void store () {

}

}

class DownloadAndStore {

void doEverything() {

new InternetDownloader().downloadFromInternet();

new DataParser().parseData();

new DatabaseStorer().storeIntoDatabase();

}

}