**Assignment G-28**

**Title:**

Write a Java program which will demonstrate a concept of cohesion and coupling of the various modules in the program.

**Objectives:**

1. To understand concept of cohesion in java language.

2. To understand concept coupling in java language.

**Learning Objectives:**

* To understand concept of cohesion in java language.
* To understand concept coupling in java language.

**Learning Outcome:**

* Define classesto show the concept of cohesion and coupling.
* Analyze working of functions.

**Theory:**

**COUPLING**

An indication of the strength of interconnections between program units.

By definition coupling is the degree to which one class has knowledge of another or in other words one class has a dependency upon another. Tight coupling occurs when a dependent concrete class contains a pointer to another concrete class that provides the required behavior and should be avoided. The problem here is that any changes to one class could impact the other and the person making the changes may be completely unaware of this and thus unknowingly break the class. So how do we avoid this scenario? We design by contract by using an interface to specify an API for other classes to use as discussed in the [OO Concepts - Interfaces](http://java5tutor.info/java/ooconcepts/interfaces.html) lesson or by using encapsulation as discussed in the [OO Concepts - Encapsulation](http://java5tutor.info/java/ooconcepts/encapsulation.html). The following code is an example of tight coupling and should be avoided:

/\*

Tight coupling example

\*/

class A {

int i;

B b = new B();

i = b.value; // No encapsulation of this variable in class B!

}

class B {

publicint value; // Should be private and be accessed through public getters and setters

}

**COHESION**

Measure of how well module fits together.

Cohesion is the degree to which components of a class belong together to fit a particular role. What we want to avoid is low cohesion where a class incorporates several different aspects. A class that tries to do many things comes with higher maintenance and lower reusability.

/\*

Low cohesion example

\*/

classAllInStaff {

voidgetStaffSalary();

voidgetStaffDetails();

voidgetStaffSalesReport();

}

/\*

High cohesion example

\*/

class Accounts {

voidgetStaffSalary();

...

}

class Personnel {

voidgetStaffDetails();

...

}

classSalesReporting {

voidgetStaffSalesReport();

...

}

Benefits of Higher Cohesion:

* Highly cohesive classes are much easier to maintain and less frequently changed.
* Such classes are more usable than others as they are designed with a well-focused purpose.

**Software Required:** java, 64 bit Fedora, eclipse IDE

**Conclusion:** This program gives us the knowledge of heap data structure.

**OUTCOME**

**Upon completion Students will be able to:**

**ELO1:**Learn cohesion in java.

**ELO2:**Learn coupling in java.

**Questions asked in university exam.**

1. What is cohesion and coupling in java
2. What is java virtual machine and what is its uses?
3. Can we override Thread's run() method?
4. What is the priority of main thread in java?? and why??