

# **Software Quality Assurance Lab (ITD14)**

## **Internal Lab Assessment**

**Submitted by :**

**Shiv Kumar**

**2016UIT2563**

**IT-2**

**8th Semester**

## **WAP to demonstrate the difference between coupling and cohesion. Use CBO and LCOM matrices**

Coupling between objects (CBO) is a count of the number of classes that are coupled to a particular class i.e. where the methods of one class call the methods or access the variables of the other.

For each field in the class, you count the methods that reference it, and then you add all of those up across all fields. You then divide that by the count of methods times the count of fields, and you subtract the result from one. So, for instance, consider the classes above.

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
class sample1{
    public :
        int a;
        int na = 1;
        int b;
        int nb = 1;
        int c;
        int nc = 1;
        static int ua;
        int nua = 2;
        int nm = 3;
        int natt = 4;
        sample1()
        {
            a=0;
            b=0;
            c=0;
        }

        static void incua(int uinc)
        {
            ua+=uinc;
        }

        static int retua()
        {
            return ua;
        }
};
```

```
class sample2{
    public :
        int x;
        int nx = 1;
        int y;
        int ny = 1;
        int z;
        int nz = 1;
        static int ub;
        int nub = 2;
        int nm = 3;
        int na = 4;
        sample2()
        {
```

```

        x=10;
        y=10;
        z=10;
    }

    static void incub(int uinc)
    {
        ub+=uinc;
    }

    static int retub()
    {
        return ub;
    }
};

class sample3{
public :
    int p;
    int np = 1;
    int q;
    int nq = 1;
    int r;
    int nr = 1;
    static int uc;
    int nuc = 1;
    int na = 4;
    int nm = 2;
    sample3()
    {
        uc=0;
        sample1 s1;
        sample2 s2;
        sample1::incua(1);
        sample2::incub(1);
        uc+=2;
        p=s1.a+s2.x;
        q=s1.b+s2.y;
        r=s1.c+s2.z;
    }

    static int retuc()
    {
        return uc;
    }
};

int sample1::ua=0;
int sample2::ub=0;
int sample3::uc=0;

int main(){
    sample1 s1;
    sample2 s2;
    sample3 s3;
    cout<<"CBO of class sample1 = "<<s1.retua()<<endl;
    cout<<"CBO of class sample2 = "<<s2.retub()<<endl;
    cout<<"CBO of class sample3 = "<<s3.retuc()<<endl;
}

```

```

        cout<<"LCOM of class sample1 = "<<((float)(s1.na+s1.nb+s1.nc+s1.nua)/
(s1.nm*s1.natt))<<endl;
        cout<<"LCOM of class sample2 = "<<((float)(s2.nx+s2.ny+s2.nz+s2.nub)/
(s2.nm*s2.na))<<endl;
        cout<<"LCOM of class sample3 = "<<((float)(s3.np+s3.nq+s3.nr+s3.nuc)/
(s3.nm*s3.na))<<endl;
    }

```

```

(base) Shivs-MacBook-Air:Software Quality Assurance championballer$ g++ internal.cpp -o internal -w
(base) Shivs-MacBook-Air:Software Quality Assurance championballer$ ./internal
CBO of class sample1 = 1
CBO of class sample2 = 1
CBO of class sample3 = 2
LCOM of class sample1 = 0.416667
LCOM of class sample2 = 0.416667
LCOM of class sample3 = 0.5

```