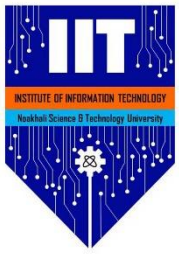


6/1/2023

# Design & Code Size of SPL-1

SE 3204 – Software Metrics

**SUBMITTED TO**

Dipok Chandra Das

Assistant Professor, IIT

Noakhali Science and Technology University

**SUBMITTERD BY**

Anupa Das Shormi (BFH1925020F)

Md.Rayhan Billah(ASH925028M)

Md.Rokonuzzaman(ASH1925018M)

# Code Size:

Program code is an integral component of software. Such code includes source code, intermediate code, byte code, and even executable code. We look at approaches for directly measuring code size. We must take great care to clarify what we are counting and how we are counting it. In particular, we must explain how each of the following is handled:

NCLOC= Non-commented line

CLOC= Comment lines of program text

Total size (LOC) = NCLOC + CLOC

Measures of the density of comments in a program.

$$\frac{\text{CLOC}}{\text{LOC}}$$

| LOC | 4071 |
|---|---|
| NCLOC | 3941 |
| CLOC | 130 |
| Average LOC | 151 |

**Number of bytes of computer storage:**

| Total code size | 201kb |
|---|---|
| Size MIN | 5kb |
| Size MAX | 35kb |
| Average Size | 12kb |

# Design Size

We can measure the size of a design in a manner similar to that used to measure code size. We will count design elements rather than LOCs. The elements that we count depend on the abstractions used to express the design, and the design aspects of interest. Thus, the appropriate size measure depends on the design methodology, the artifacts developed, and the level of abstraction. Thus, we will measure size in terms of packages, design patterns, classes, interfaces, abstract classes, operations, and methods.

**Packages:** Number of sub packages, number of classes, interfaces (Java), or abstract classes

**Design patterns:**

- Number of different design patterns used in a design
- Number of design pattern realizations for each pattern type
- Number of classes, interfaces, or abstract classes that play roles in each pattern realization.

**Classes, interfaces, or abstract classes:** Number of public methods or operations, number of attributes

**Methods or operations:** Number of parameters, number of overloaded versions of a method or Operation. One of the earliest object-oriented design size measures was the weighted methods per class. (WMC) measure Calculate the weight of a method using cyclomatic complexity and finding out the weight of method for a class.

In Below we measure design size of our project Chemistry Calculator

| Class name | Public method | Overloaded Method | No of attribute | No of parameter |
|---|---|---|---|---|
| Titrationpanel | 1 | | 30 | |
| Concentrationpanel | 1 | | 29 | |
| Equation balance panel | 1 | | 20 | |
| Sidebar | 3 | | 13 | 3 |
| Electron config panel | 1 | | 15 | |
| Fraction | 15 | 2 | 2 | 15 |
| Molar mass panel | 1 | | 17 | |
| Parcent of complition | 1 | | 13 | |
| Need help panel | 1 | | 14 | |
| Compound manager | 10 | 6 | 3 | 23 |
| Atom | 8 | | 5 | 2 |
| Home | 2 | | 10 | |

| | | | | |
|---|---|---|---|---|
| History frame | 2 | | 8 | |
| Matrix | 6 | 1 | 2 | 2 |
| Compound | 5 | | 4 | 1 |
| FixPieChart | 1 | | 5 | 1 |
| Concentration | 4 | | 3 | 3 |
| Titration | 2 | | 3 | 6 |
| Converter | 3 | | 1 | 6 |
| Equation Balancer | 2 | | 2 | 2 |
| Database | 2 | | 1 | |
| Formater | 2 | | | 1 |
| DatabaseSerializer | 2 | | | |
| InsufficientDataException | 1 | | | |
| InvalidEquationException | 1 | | | 1 |
| InvalidAtomException | 1 | | | 1 |
| History | 1 | | | 1 |

| | |
|---|---|
| Number of Package | 3 |
| Number of Sub-packages | 0 |
| Number of Class | 29 |
| Number of Interface | 1 |
| Number of Abstract Class | 0 |
| Number of Methods | 142 |
| Number of Public Method | 80 |
| Number of Overloaded Method | 9 |
| WMC | 6 |
| Number of Design Pattern | 3 |
| Number of attributes | 200 |
| Number of Parameter | 68 |
| Density Of Comments | 3% |