**Objects and Classes**  
**Workshop 3**

**In this workshop, you’ll learn:**

* Design and code a class for a simple object,
* Write code that includes robust user input validation, and
* Write code that includes formatted program output.

**1. Fractions**

Design and code a class named Fraction that holds information about a single fraction.  Place your class declaration and definition in a file named Fraction.java

The Fraction class has the following members:

**Data/Attributes/Fields**

* **numerator**: an integer holding the numerator.
* **denominator:** an integer holding the denominator.

**Behaviors/Operations/Methods**

* **Default constructor:** leave the object in safe empty state.
* **Constructor:** to set values for attributes.
* **String toString()**: displays the fraction information on standard output in fraction format, for example 2/3
* **void reduce()**: reduce the current fraction
* **void add(Fraction a, Fraction b)**: add two fractions and return the current fraction
* **void subtract(Fraction a, Fraction b)**: subtract two fractions and return the current fraction
* **void multiply(Fraction a, Fraction b)**: multiply two fractions and return the current fraction
* **void divide(Fraction a, Fraction b)**: divide two fractions and return the current fraction

Design and code a main program that accepts information for arbitrarily given fractions,   
adds, subtracts, multiplíe, divides two fractions and displays the resulting fraction information.

**Additional proposed exercises:**

**2. Atoms**

Design and code a class named Atom that holds information about a single atom.  Place your class declaration and definition in a file named Atom.java

The Atom class has the following members:

**Data/Attributes/Fields**

* **number**: an integer holding the atomic number,
* **symbol**: a string that holding the atomic symbol,
* **fullname:** a string that holding the full name of the atom
* **weight**: a floating-point value holding the atomic weight.

**Behaviors/Operations/Methods**

* **Default constructor:** leave the object in safe empty state.
* **void accept()**: prompts for and accepts from standard input
* **void display()**: displays the atomic information on standard output.

Design and code a main program that accepts information for up to 10 atomic elements and displays the atomic information in tabular format.

The program output might look something like:

|  |
| --- |
| Atomic Information  ==================  Enter atomic number : 3  Enter symbol : Li  Enter full name : lithium  Enter atomic weight : 6.941  Enter atomic number : 20  Enter symbol : Ca  Enter full name : calcium  Enter atomic weight : 40.078  Enter atomic number : 30  Enter symbol : Zn  Enter full name : zinc  Enter atomic weight : 65.409  Enter atomic number : 0    3 Li lithium 6.941  20 Ca calcium 40.078  30 Zn zinc 65.409 |