SIM GLOBAL EDUCATION UOW AUSTRALIA

SCIT

School of Computing and Information Technology
Faculty of Engineering & Information Sciences

CSIT101
Object Oriented Design and Programming
Assignment 1

Objectives:

Practice java programming with classes and objects, constructors, copy constructors, enum type, array, `ArrayList` (generic version), overloaded methods, passing by reference etc.

Task (7 marks)

In mathematics, a set is a collection of *distinct* elements and elements in a set are not in order. Here are some examples of sets:

(a) A set of integers, e.g. integerSet = {3, 1, 4, 2}
(b) A set of fruits, e.g. fruitSet = {apple, orange, papaya}
(c) A set of characters, e.g. charSet = {'A', 'm', '&'}

We always enclose elements of sets inside a pair of { }.

Here are some other properties on sets:

(1) A set can be empty, i.e. no element. We call it empty set. In mathematics, we have a special symbol to denote empty set. Convenient to our design later, we will use { } to denote an empty set.

(2) When checking an element is inside a set. We call it "belong to".

(3) If a set contains in another set, we call it subset. For example, {1, 2, 3} is a subset of {2, 3, 4, 6, 1}. Therefore, empty set is a subset of every set.

(4) The *cardinal number* of a set is the number of elements in a set.

(5) The union of two sets A and B are all the elements belong to A and B, minus the duplications. For example, if A = {1, 2, 3} and B = {2, 3, 4, 5}, the union is {1, 2, 3, 4, 5}.

(6) The intersection of two sets A and B are the common elements of A and B. Using the example quoted in (5), the intersection is {2, 3}.

(7) The equality of two sets A and B are all the elements of A are in B and all the elements of B are in A. Or alternatively, A is the subset of B and vice versa.

(8) The difference of two sets A and B, for example A – B, is those elements in B should not be in A. For example, if A = {1, 2, 3} and B = {2, 3, 4, 5}, A – B = {1} and B – A = {4, 5}.

We have all the required properties for our task.

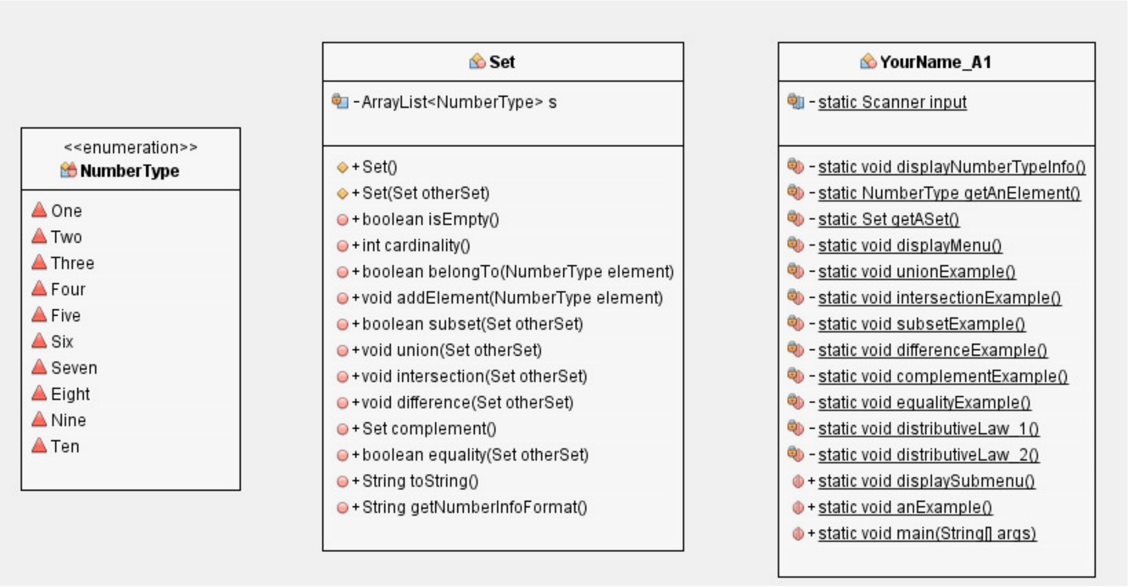
In this assignment, our ***universal set*** is the 10 enumeration constants of numbers:

One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Ten.

Data validation is not necessary in this design, when I want to add in a number, you can assume that this is a valid number in the enumeration constants; i.e. we have restricted ourselves in a subset of the above 10 numbers.

The best way to test your design is to develop an educational system to teach some basic set theory. All the sets used in your design should be *randomly generated* by the system i.e., the sizes (also known as cardinal number, from 0 to 10 elements) and the elements.

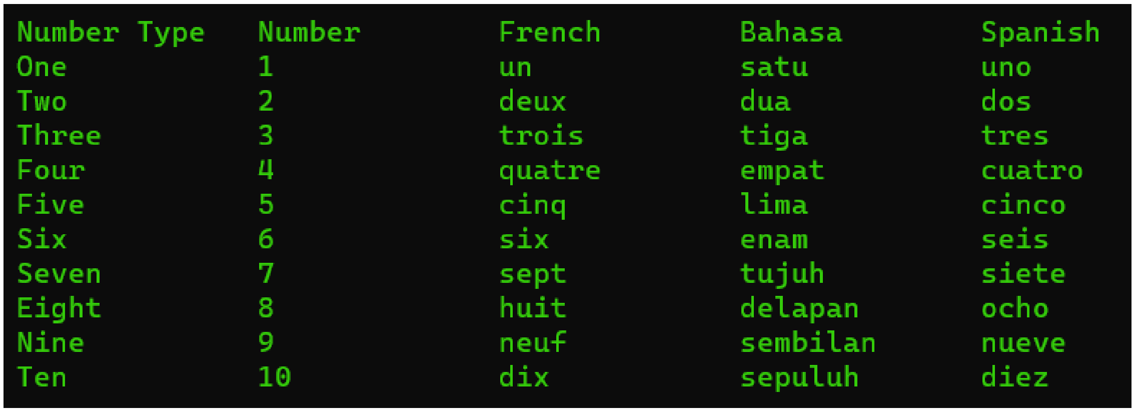
Let us explore the following UML diagram for the whole task:


<<enumeration>>
NumberType
One
Two
Three
Four
Five
Six
Seven
Eight
Nine
Ten
Set
-ArrayList<NumberType> s
+Set()
+Set(Set otherSet)
+boolean isEmpty()
+int cardinality()
+boolean belongTo(NumberType element)
+void addElement(NumberType element)
+boolean subset(Set otherSet)
+void union(Set otherSet)
+void intersection(Set otherSet)
+void difference(Set otherSet)
+Set complement()
+boolean equality(Set otherSet)
+String toString()
+String getNumberInfoFormat()
YourName_A1
-static Scanner input
-static void displayNumberTypeInfo()
-static NumberType getAnElement()
-static Set getASet()
-static void displayMenu()
-static void unionExample()
-static void intersectionExample()
-static void subsetExample()
-static void differenceExample()
-static void complementExample()
-static void equalityExample()
-static void distributiveLaw_1()
-static void distributiveLaw_2()
+static void displaySubmenu()
+static void anExample()
+static void main(String[] args)

Let us look at each of the classes:

(a) Enumeration class **`NumberType`**

This is the universal set consists of 10 constants (One to Ten) that a subset is constructed. Each enum constant has a few descriptions: Arabic numbers (1, 2, 3…), French numbers, Malay / Bahasa, Spanish … Feel free to change to some other languages, but I may not have decoder to interpret your mother tongue.


```
Number Type   Number        French        Bahasa        Spanish
One           1             un            satu          uno
Two           2             deux          dua           dos
Three         3             trois         tiga          tres
Four          4             quatre        empat         cuatro
Five          5             cinq          lima          cinco
Six           6             six           enam          seis
Seven         7             sept          tujuh         siete
Eight         8             huit          delapan       ocho
Nine          9             neuf          sembilan      nueve
Ten           10            dix           sepuluh       diez
```

(b) Class **`Set`**

We use an array list to represent a set, which is an instance variable defined inside the class. The normal set operations: belong to, contains, union, intersection, complement, difference, subset, equality is some of the set's operations. The `toString` method returns a String of some enum constants (can be empty) enclosed between braces and the `getEnumFormat` method returns its equivalent to one of the descriptions. You will see in more detail later when work on one of the set operations.

You should use the default constructor to construct an empty set. Do some deep copying in the copy constructor; you may need this constructor to perform some of the subtasks.

Only the following methods can be used from the ArrayList:

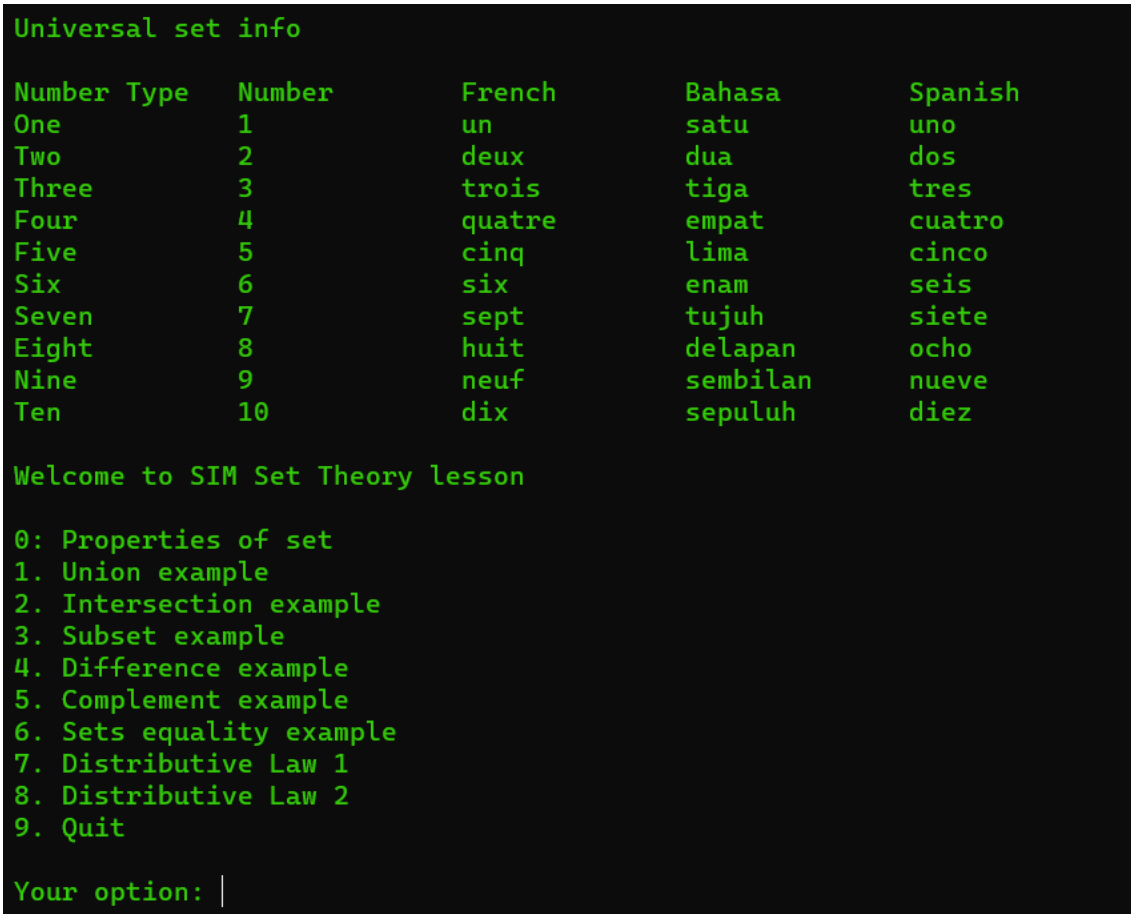
add, get, contains, remove, and size.

You should fully explore the set operations designed by you once you have implemented them. In the whole design, though you use an array list to denote a set, you should not perform too many operations on array list, you should assume that this list is "implicitly" defined

(c) Main class

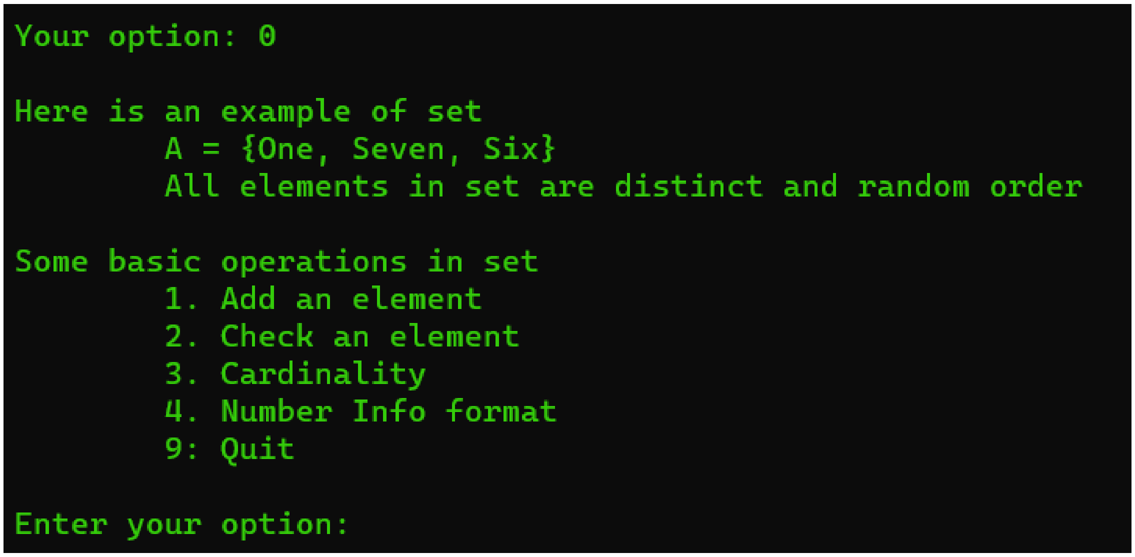
We are now ready to present the whole system. You are required to design an educational system to teach basic set theory. We propose the following interactions for your system:

When you execute your program, the system will display the info for the universal set (a call to `displayNumberTypeInfo` method, *display once only*) and followed by a menu (the display of menu is repeated after an operation)



```
Universal set info

Number Type   Number        French        Bahasa        Spanish
One           1             un            satu          uno
Two           2             deux          dua           dos
Three         3             trois         tiga          tres
Four          4             quatre        empat         cuatro
Five          5             cinq          lima          cinco
Six           6             six           enam          seis
Seven         7             sept          tujuh         siete
Eight         8             huit          delapan       ocho
Nine          9             neuf          sembilan      nueve
Ten           10            dix           sepuluh       diez

Welcome to SIM Set Theory lesson

0: Properties of set
1. Union example
2. Intersection example
3. Subset example
4. Difference example
5. Complement example
6. Sets equality example
7. Distributive Law 1
8. Distributive Law 2
9. Quit

Your option: 
```

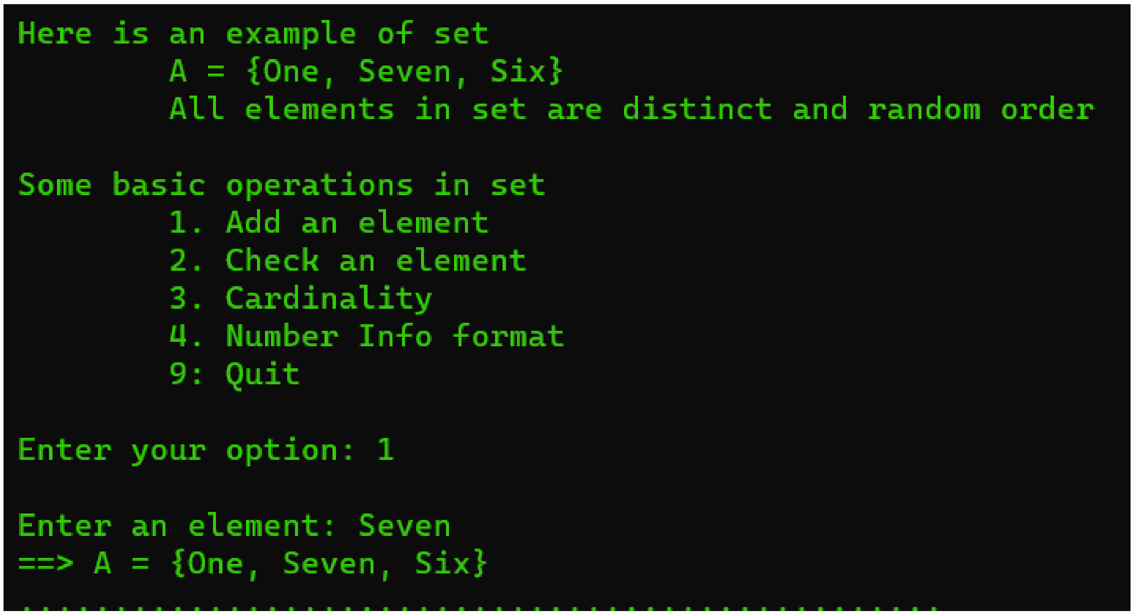
When you enter option 0, you will see the following interactions:



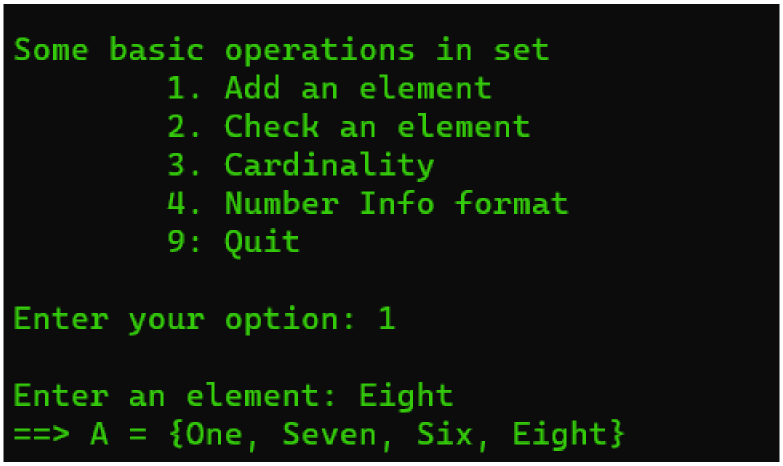
```
Your option: 0

Here is an example of set
        A = {One, Seven, Six}
        All elements in set are distinct and random order

Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Number Info format
        9: Quit

Enter your option:
```

A set is generated, and a submenu is displayed. You can now try a few simple set operations, add an element, belong to operation, display the cardinal number and a display in enum format. *Note that the sub-menu will be repeated after each operation.*

Let us enter the option 1 in the submenu,



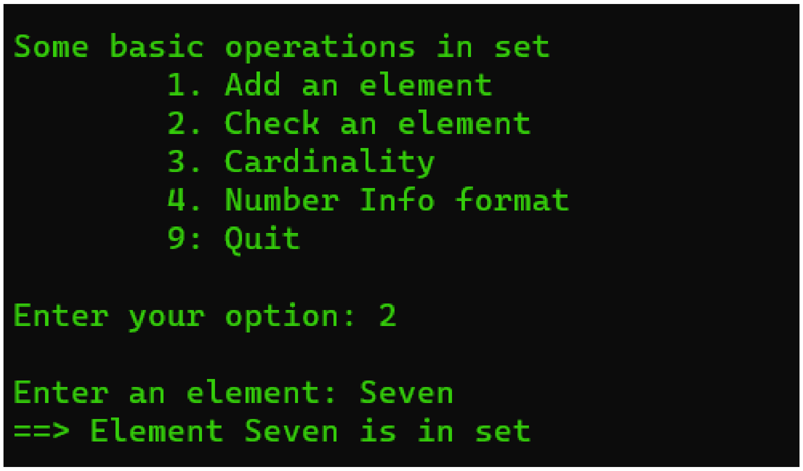
```
Here is an example of set
        A = {One, Seven, Six}
        All elements in set are distinct and random order

Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Number Info format
        9: Quit

Enter your option: 1

Enter an element: Seven
==> A = {One, Seven, Six}
```



```
Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Number Info format
        9: Quit

Enter your option: 1

Enter an element: Eight
==> A = {One, Seven, Six, Eight}
```

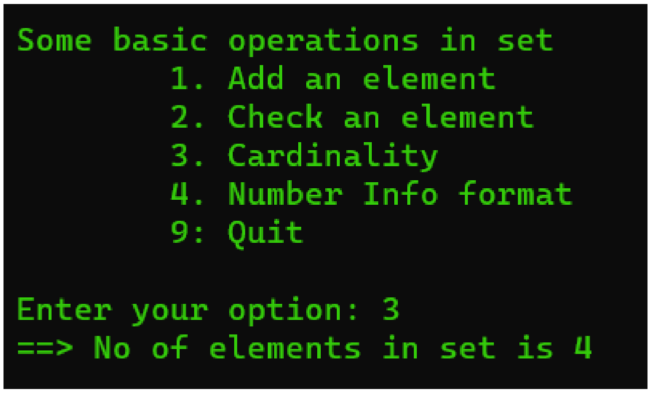
You can see in the above interactions, adding an element which is already inside the set, the final set remains unchanged; otherwise, this distinct element is added to the set.

Let us explore option 2



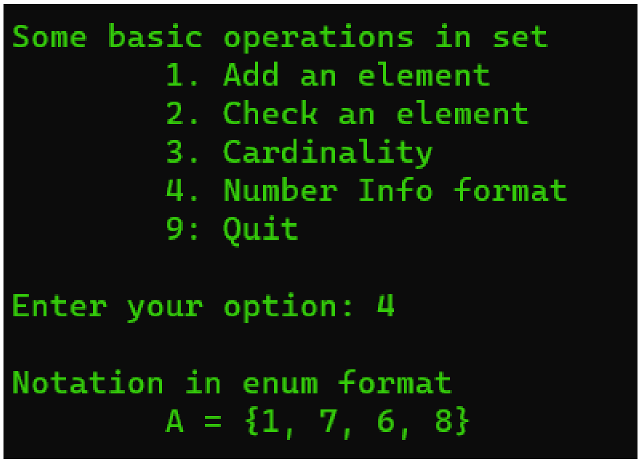
```
Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Number Info format
        9: Quit

Enter your option: 2

Enter an element: Seven
==> Element Seven is in set
```

For option 3, the system just simply displays the cardinal number:



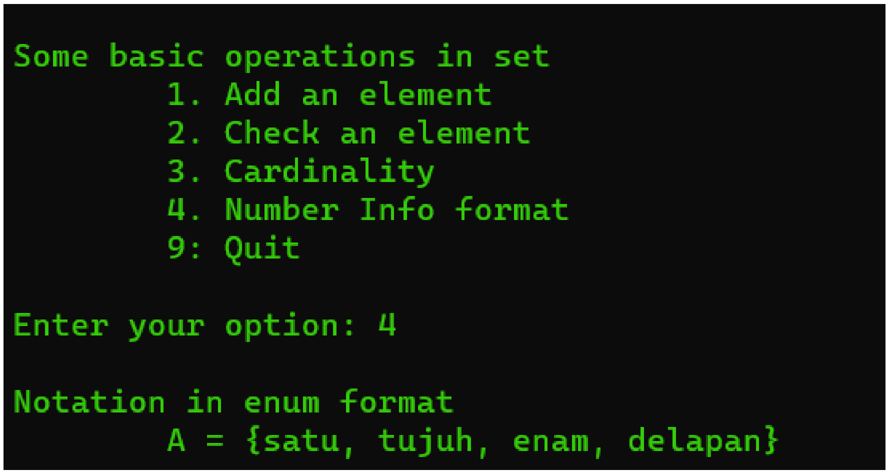
```
Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Number Info format
        9: Quit

Enter your option: 3
==> No of elements in set is 4
```

For option = 4: the system invokes another format (to display one of the enum descriptions) to display the corresponding set's information:



```
Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Number Info format
        9: Quit

Enter your option: 4

Notation in enum format
        A = {1, 7, 6, 8}
```

Let us try option = 4 one more time; a different random format is displayed:



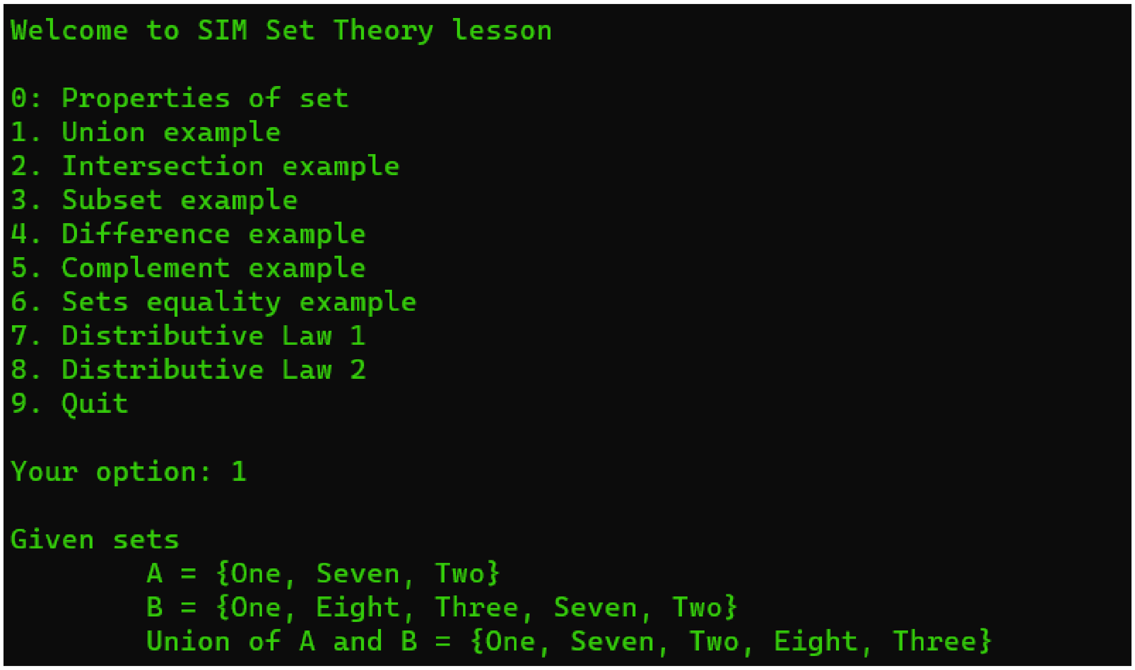
```
Some basic operations in set
        1. Add an element
        2. Check an element
        3. Cardinality
        4. Number Info format
        9: Quit

Enter your option: 4

Notation in enum format
        A = {satu, tujuh, enam, delapan}
```

You can continue to stay in the submenu or enter 9 to go back to the main menu.

Let us choose 9 to go back to the main screen to test other operations.

Let us now explore each of the options in the main menu:

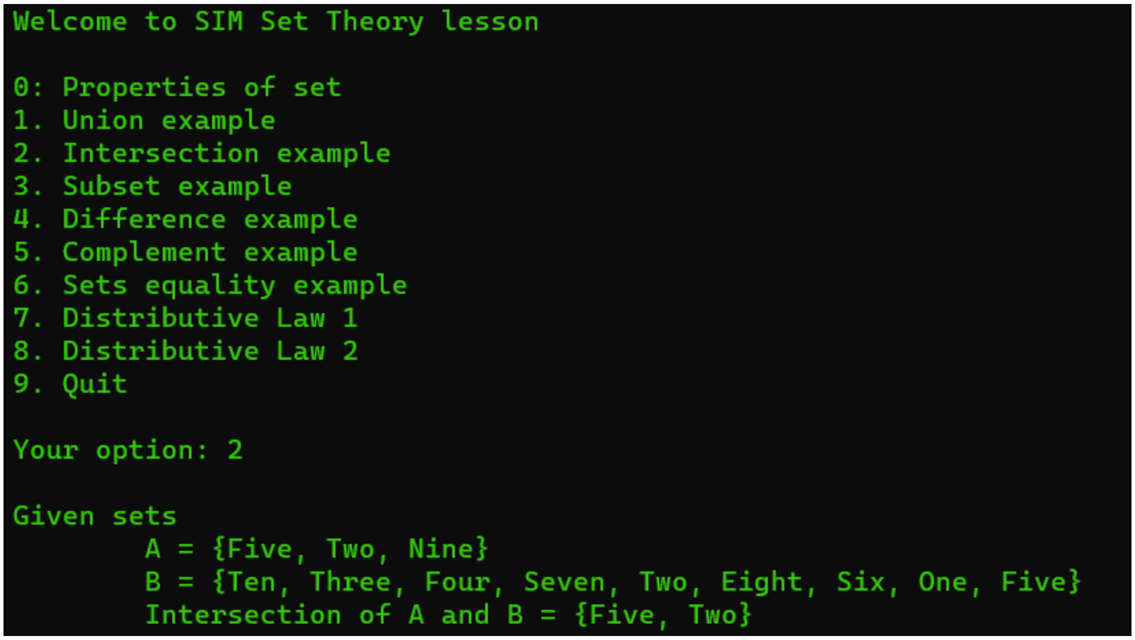
In the main menu, you choose option 1:



```
Welcome to SIM Set Theory lesson

0: Properties of set
1. Union example
2. Intersection example
3. Subset example
4. Difference example
5. Complement example
6. Sets equality example
7. Distributive Law 1
8. Distributive Law 2
9. Quit

Your option: 1

Given sets
        A = {One, Seven, Two}
        B = {One, Eight, Three, Seven, Two}
        Union of A and B = {One, Seven, Two, Eight, Three}
```

In option 1, the system randomly generates two sets and displays the union of these two sets.

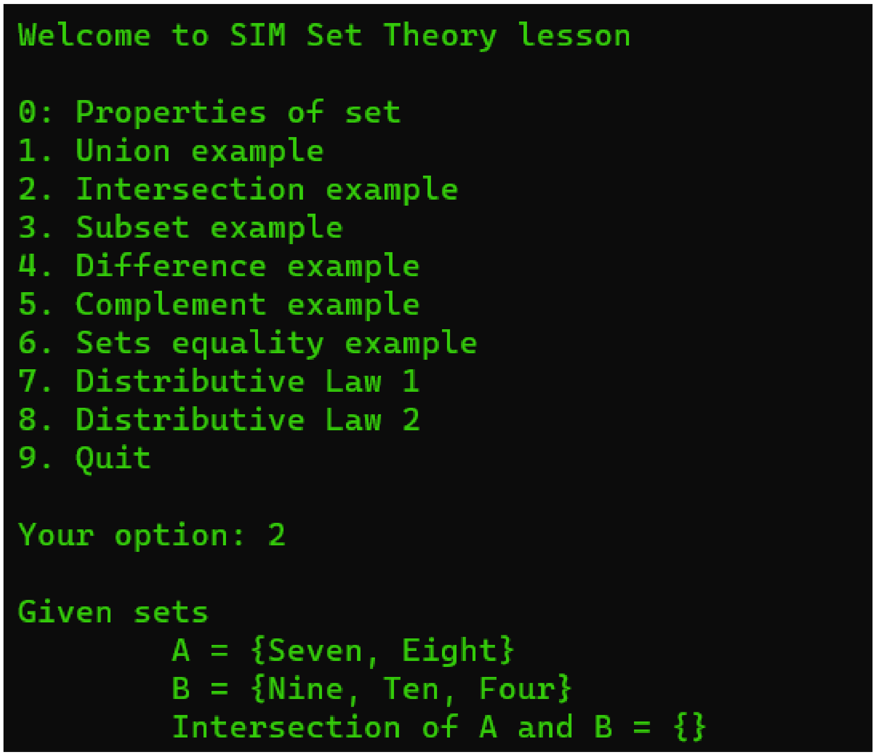
The same is done for option 2, but evaluate the intersection of the two sets:



```
Welcome to SIM Set Theory lesson

0: Properties of set
1. Union example
2. Intersection example
3. Subset example
4. Difference example
5. Complement example
6. Sets equality example
7. Distributive Law 1
8. Distributive Law 2
9. Quit

Your option: 2

Given sets
        A = {Five, Two, Nine}
        B = {Ten, Three, Four, Seven, Two, Eight, Six, One, Five}
        Intersection of A and B = {Five, Two}
```

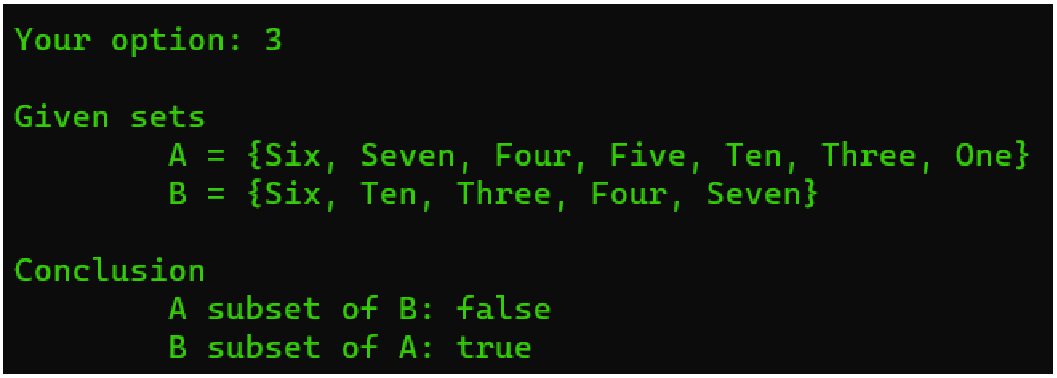
Important to note, the main menu is always displayed.

In the following screen shot, you see the notation of an empty set.



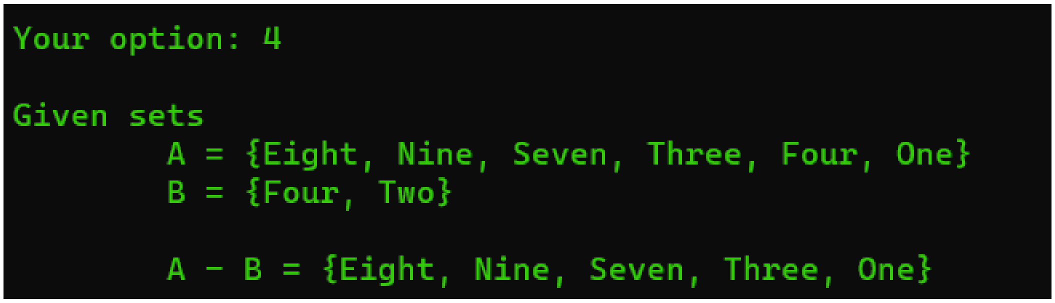
```
Welcome to SIM Set Theory lesson

0: Properties of set
1. Union example
2. Intersection example
3. Subset example
4. Difference example
5. Complement example
6. Sets equality example
7. Distributive Law 1
8. Distributive Law 2
9. Quit

Your option: 2

Given sets
        A = {Seven, Eight}
        B = {Nine, Ten, Four}
        Intersection of A and B = {}
```

For option 3, the subset operation:



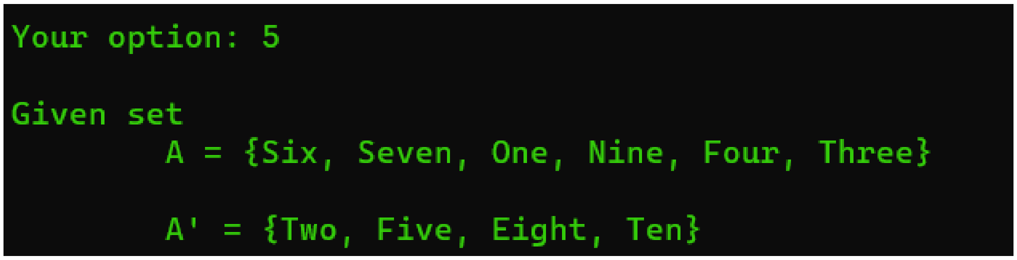
```
Your option: 3

Given sets
        A = {Six, Seven, Four, Five, Ten, Three, One}
        B = {Six, Ten, Three, Four, Seven}

Conclusion
        A subset of B: false
        B subset of A: true
```

For option 4, the difference of two sets:



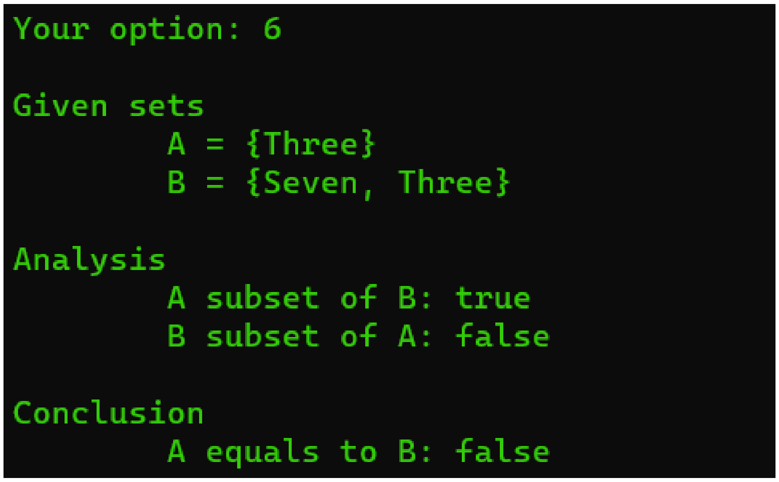
```
Your option: 4

Given sets
        A = {Eight, Nine, Seven, Three, Four, One}
        B = {Four, Two}

        A - B = {Eight, Nine, Seven, Three, One}
```

Now, option 5, the complement of a set is done with the universal set. Our universal set is the set of numbers. The following shows some of the interactions and displays:

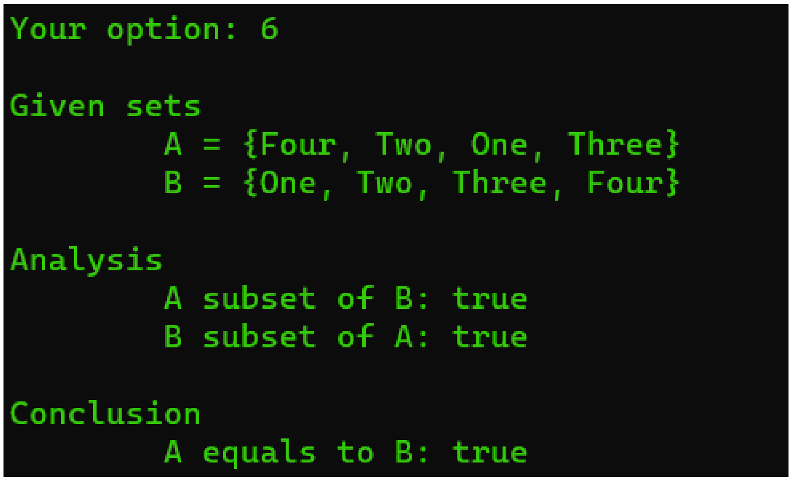


```
Your option: 5

Given set
        A = {Six, Seven, One, Nine, Four, Three}

        A' = {Two, Five, Eight, Ten}
```

Option 6 is the set equality. The following shows some of the possible interactions and display:



```
Your option: 6

Given sets
        A = {Three}
        B = {Seven, Three}

Analysis
        A subset of B: true
        B subset of A: false

Conclusion
        A equals to B: false
```



```
Your option: 6

Given sets
        A = {Four, Two, One, Three}
        B = {One, Two, Three, Four}

Analysis
        A subset of B: true
        B subset of A: true

Conclusion
        A equals to B: true
```

Option 7 is for first distributive law:

Distributive Law states that, the sum and product remain the same value even when the order of the elements is altered.

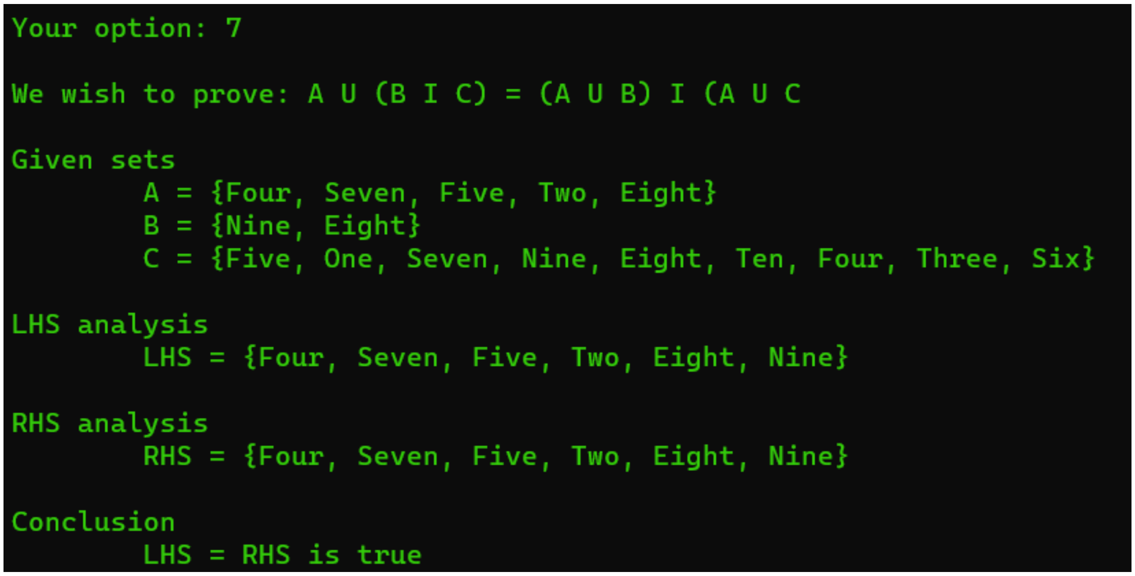
First Law: A ∪ (B ∩ C) = (A ∪ B) ∩ (A ∪ C)

We use symbol "I" for intersection in display:

Let us explain the 1st law

We first compute the expression on the left-hand side. To do this, you need to compute (B ∩ C) and then A ∪ (B ∩ C).

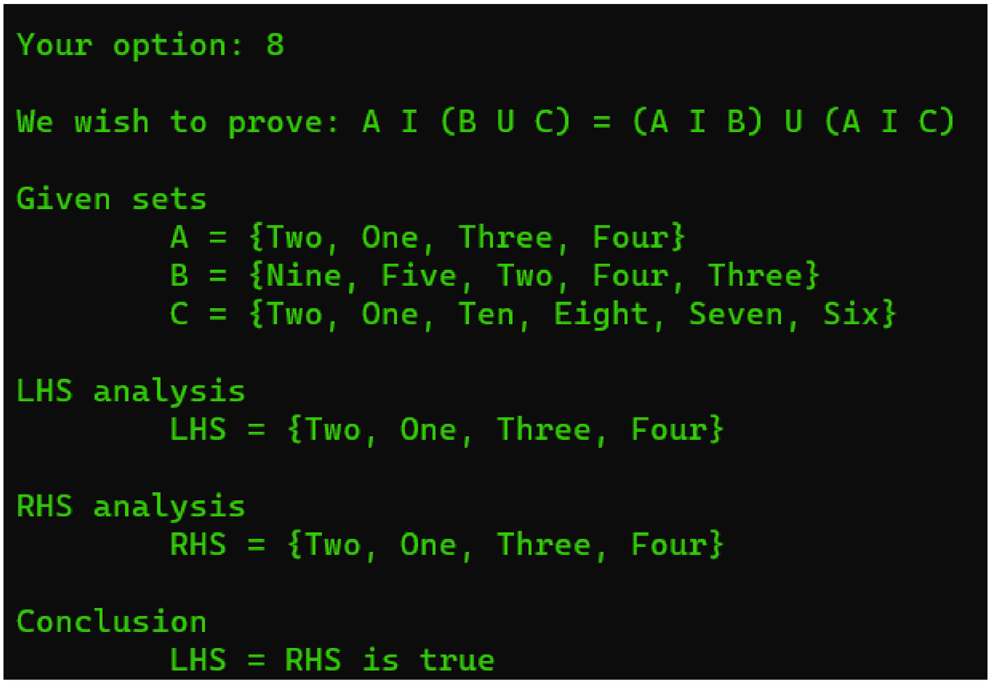
To compute the expression on the right-hand side, you need to compute (A ∪ B), (A ∪ C) and then (A ∪ B) ∩ (A ∪ C).



```
Your option: 7

We wish to prove: A U (B I C) = (A U B) I (A U C

Given sets
        A = {Four, Seven, Five, Two, Eight}
        B = {Nine, Eight}
        C = {Five, One, Seven, Nine, Eight, Ten, Four, Three, Six}

LHS analysis
        LHS = {Four, Seven, Five, Two, Eight, Nine}

RHS analysis
        RHS = {Four, Seven, Five, Two, Eight, Nine}

Conclusion
        LHS = RHS is true
```

Option 8 is for second distributive law:

Second Law: A ∩ (B ∪ C) = (A ∩ B) ∪ (A ∩ C)



```
Your option: 8

We wish to prove: A I (B U C) = (A I B) U (A I C)

Given sets
        A = {Two, One, Three, Four}
        B = {Nine, Five, Two, Four, Three}
        C = {Two, One, Ten, Eight, Seven, Six}

LHS analysis
        LHS = {Two, One, Three, Four}

RHS analysis
        RHS = {Two, One, Three, Four}

Conclusion
        LHS = RHS is true
```

IMPORTANT TO NOTE

Put all your classes in a file called **`YourName_A1.java`** and make sure that this file can be compiled and can be executed. Upload **ONLY** this file to Moodle**. ALL ZIP FILE SUBMISSION WILL BE REJECTED**

NOTE THAT ALL CLASSES SHOULD NOT BE PUBLIC!!!!!

No re-submission will be allowed after grading.

In the above file, remember to put down your name and the following declaration (some similar contents):

// Tell me if it is your own work, and whether you have passed your
// program to your friends etc
// and willing to accept whatever penalty given to you.

- **Wrong file name -0.5 mark**
- **No declaration, no name etc -0.5 mark**
- **Failing to demo -1 mark**
- **Programs' indentations and alignment of statements -0.5 mark**