



Objective:

- Understanding and implementing weak/strong Aggregation and dealing dynamic memory allocation for objects.
- Reusability through composition/aggregation.

Task 1: Set using Array

You must be quite familiar with the class 'Array' implemented below. The other class i.e. 'Set' is also not new to you. Although the purpose of class 'Set' is same like the previous version but its data members are changed and 'Set' is implemented using class 'Array' object. I have defined a few functions in Set class, you are required to implement rest of the function given in Set class. Remember: You must not change anything private/public in class Array. Wisely decide about the fate of destructor and copy constructor of class 'Set'

```
class Array
{ //Discussed in class };

class Set
{
    Array data;
    int noOfElements;
public:
    Set( int cap = 0 ):data(cap)
    { noOfElements=0; }
    void insert ( int element );
    void remove ( int element );
    void print ( ) const;
    int getCardinality() const;
    bool isMember ( int val ) const;
    int isSubSet ( Set s2 ) const;
        //return 1 if *this is subset of s2.
        //return -1 if s2 is subset of *this.
        //return 0 if no one is subset.
        //return 2 if it is improper subset.
    void reSize ( int newcapacity );
    Set calcUnion ( const Set & s2 ) const;
    Set calcIntersection ( const Set & s2 ) const;
    Set calcDifference ( const Set & s2 ) const;
    Set calcSymmetricDifference ( const Set & s2 ) const;
    void displayPowerSet ( ) const;
};
```

Task 2: → PF >= 65 : Bravo to others if they do it too.

Complete the following class which will convert the floating-point value into 32b as per IEEE754. You got to use the standard/IEEE754 algorithm/method of conversion as discussed in lab/lecture. No short cuts allowed.

<pre>class IEEE754 { BitArray floatingBits; float value = 1; public: IEEE754(); void updateFloatingPointValue(float); void updateIEEE754FloatingBits(BitArray); float getValue(); BitArray getIEEE754FloatingBits(); };</pre>	<pre>int main() { IEEE754 fp; fp.getValue();//display 1; fp.getIEEE754FloatingBits().dump(); //display 00111111 10000000 00000000 00000000 fp.updateFloatingPointValue(12.375); fp.getIEEE754FloatingBits().dump(); //display 01000001 01000110 00000000 00000000 return 0; }</pre>
---	---