

### Assignment 2 part 2

<b>Deadline:</b>	<b>Anytime before</b> Sunday 8 October 2017, time due 23:59
<b>Evaluation:</b>	10 marks – which is 5% of your final grade
<b>Late Submission:</b>	5% per hour (or fraction of hour) it is late
<b>Teams</b>	The assignment can be done individually or in pairs (of at most 2 students)
<b>Purpose:</b>	Practice with C++ exceptions and templates

**Problem to solve:** You will write a class template to model a simplified mathematical-like set<sup>1</sup> type.

**Requirements:** You will design, implement and test a class template called Set that will be able to compute union and intersection of sets as + and \* operators. For example  $\{4,2,3\} + \{9,4,8,2\}$  will be  $\{8,3,9,2,4\}$ , i.e. the union of the two sets and  $\{3,2,4\} * \{8,4,9,2\}$  will be  $\{4,2\}$  i.e. the intersection of the two sets.

Please note that it is not required to display the elements in sorted order. Your solution will have to implement and use at least two exception classes to handle possible run-time error cases.

All declarations & definitions for the above mentioned classes must be written in a file called **a2p2.h**; do not include a main function in the a2p2.h file, please.

Here is the skeleton of the file a2p2.h:

```
#ifndef A2P2_H
#define A2P2_H
#include <exception>
#include <iostream>
using namespace std;

//=====part a-Comments here:

//=====part b-author's details
void info(){/* missing code */}

//=====part c-exception classes:
class RemoveFromEmpty : exception{
public:
    RemoveFromEmpty(){/* missing code */}
    const char* what() const noexcept {/* missing code */}
private:
    string mMessage;
};
class NonExistingElem:exception{
    /* to be thrown when the element to be removed is not found in
    the set -----code missing */
};

//=====part d-Set class template
template <typename EType>
class Set{
public:
    //constructors
    Set( );
    Set( const Set & rhs );
    Set( Set && rhs );
    //destructor
```

<sup>1</sup> Note that this set type is designed as an assignment exercise to practice using templates. In real life for all cases where a set type is needed the **STL-set** class template should be used.

```

~Set( );
//operators overloaded
Set & operator=( const Set & rhs );
Set & operator=( Set && rhs );
Set operator+( const Set & rhs ) const; //set union
Set operator*( const Set & rhs ) const; //set intersection

//methods
bool isElement( const EType & x ) const;
bool isEmpty( ) const;
int getSize( ) const;
//display on out all elements in the set between {...}
void print( ostream & out = cout ) const;

void setToEmptySet( );

//methods to work with individual elements of a set
void insert( const EType & x );
void remove( const EType & x );

private:
    struct Node{// type of the elements of the set
        EType mData;
        Node *mNext;
        Node( const EType & d = EType( ), Node *n = nullptr )
            : mData( d ), mNext( n ) { }
    };
    Node *mFirst;
    int mSize; // to have an efficient getSize().
};

//Write the definitions of all Set function members here:

//=====part e-the output operator:
template <typename EType>
ostream & operator<< /* code missing here */
#endif

```

Your a2p2.h file should be organized as follows:

- a) comments with names of all authors of the code solution, assignment number, etc...
- b) the definition of the function **info( )**, that should display the names of the authors of the code solution
- c) two simple exception classes derived from `std::exception` class:
  - i) `EmptySetException` to be thrown when attempting to remove an element from an empty set
  - ii) `NonExistingElem` to be thrown when the element to be removed is not found in the set
- d) the class template `Set` declaration followed by the implementation of all member functions of the `Set` class
- e) the implementation of the overloading of the output operator for the `Set` type implemented as a non-friend function of `Set` type

### Important notes:

- a) Update all comments in the file you send for marking
- b) You may add more data members or member functions to the `Set` class if you think it will be useful but your solution **should not use any of the STL containers** for this assignment and should use/work with the code of the `Set` class from the skeleton above. In case you add new

data/function members to the Set class you must provide comments explaining the motivations for your additions.

c) In case you remove/comment out any code from the Set class skeleton or include a main function in your h file, the total mark for this assignment will be 0.

**Hand-in:** Submit a2p2.h electronically using STREAM.

### Testing

To help you with testing your solution I include here a possible main function and the output produced. As it happens for all 159.234 assignments please be aware that we might include more tests in the main function when we mark your solution.

```
#include <iostream>
#include "a2p2.h"
using namespace std;

//It works only for sets of integers
template <typename T = int>
void testCopyCtr(Set<T> st){
    cout<< "func <<": ";
    st.insert(23); //because of this statement
    st.print();
    cout<<endl;
}

int main( ){
    info(); //authors details
    try{
        Set<int> s1, s2;
        s1.insert( 8 );
        s1.insert( 3 );
        s1.insert( 1 );
        s1.insert( 4 );
        s1.insert( 1 );
        s1.remove( 8 );
        s1.insert( 2 );
        s2.insert( 4 );
        s2.insert( 2 );
        s2.insert( 6 );
        cout << "S1: " << s1 << ", size= " << s1.getSize( ) << endl;
        cout << "S2: " << s2 << ", size= " << s2.getSize( ) << endl;

        Set<int> s3 = s1+ s2; //union
        Set<int> s4 = s1* s2; //intersection
        cout << "s1 + s2: " << s3 << endl;
        cout << "s1 * s2: " << s4 << endl;

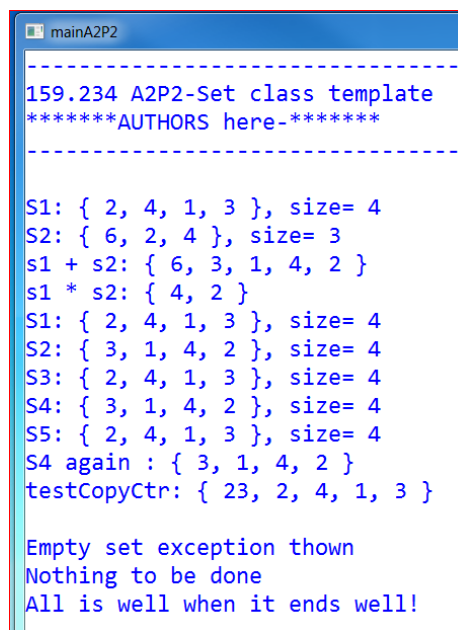
        Set<int> s5 = s4 = s3 = s2 = s1 =s1;
        cout << "S1: " << s1 << ", size= " << s1.getSize( ) << endl;
        cout << "S2: " << s2 << ", size= " << s2.getSize( ) << endl;
        cout << "S3: " << s3 << ", size= " << s3.getSize( ) << endl;
        cout << "S4: " << s4 << ", size= " << s4.getSize( ) << endl;
        cout << "S5: " << s5 << ", size= " << s5.getSize( ) << endl;

        cout << "S4 again : " << s4 << endl;
        testCopyCtr(s4);
```

```

        Set<float> sf;
        sf.remove(3);
    }
    catch(RemoveFromEmpty ex) {
        cout<<endl<<ex.what()<<endl;
        cout<<"Nothing to be done\n";
    }
    catch(NonExistingElem ex) {
        cout<<endl<<ex.what()<<endl;
        cout<<"Nothing to be done\n";
    }
    cout<<"All is well when it ends well!\n";
    return 0;
}

```



```

-----
159.234 A2P2-Set class template
*****AUTHORS here-*****
-----

S1: { 2, 4, 1, 3 }, size= 4
S2: { 6, 2, 4 }, size= 3
s1 + s2: { 6, 3, 1, 4, 2 }
s1 * s2: { 4, 2 }
S1: { 2, 4, 1, 3 }, size= 4
S2: { 3, 1, 4, 2 }, size= 4
S3: { 2, 4, 1, 3 }, size= 4
S4: { 3, 1, 4, 2 }, size= 4
S5: { 2, 4, 1, 3 }, size= 4
S4 again : { 3, 1, 4, 2 }
testCopyCtr: { 23, 2, 4, 1, 3 }

Empty set exception thrown
Nothing to be done
All is well when it ends well!

```

### Miscellaneous:

1. The program must be your own work. Please be aware that you might be asked to explain to your lecturer how your program works. **If you cannot explain it, then it is not yours and you will get 0 marks for that assignment.** Attributing someone else's work as your own is plagiarism, and it is a violation of Massey University policy. We might file an official complaint against any student who we believe has committed plagiarism.
2. Marks will be allocated for: correctness, completeness, use of C++ constructs presented in lectures/tutorials, simple and clear solution, good documentation, and structured output display (on screen).
3. Using goto, **non-constant global variables** or C-like I/O constructs (i.e printf, fprintf, scanf, FILE\*, etc) is not allowed and it will be penalised. **Only const global variables are allowed.**
4. Programs that **do not run or do not compile in the (Albany) labs, using gcc(SciTe), get 0 marks.**
5. Suspicious **similar solutions** will **all** get 0 marks-see also point 1 above.
6. Write YOUR ID NUMBER(S), and YOUR **FAMILY** NAME(S) first, assignment number, what the program does at the beginning of the file you send electronically and at least comment each function.
7. When working in pair, **send one solution file per team.**
8. The assignment will be previewed on Thursday lecture before the assignment is due.

**If you have any questions about this assignment, please ask the lecturer before its due time!**