

# 159.234 Assignment 2-part 2

Deadline:	Anytime before: Sunday 17 May 2015, time due 00:00 (midnight)
Evaluation:	10 marks – which is 5% of your final grade
Late submission	1 mark deduction for every 8 hours it is late
Purpose:	Practice with C++ dynamic memory management and templates

#### Problem to solve:

In this assignment, you will create a class template VecT generalizing the VectType required for Assignment 1 Part 2. The template class VecT should provide all behaviour provided by VectType (from Assignment 1 Part2) and two extra member functions:

- a move constructor and
- a move assignment.

### You should:

- overload the input and output operators as global function (not as friend functions of the template class VecT),
- define and implement the class VecT,
- write all the required code in a single file named A2P2.h that you should send for marking via Stream- do not include a main function in the file A2P2.h
- write good comments to explain the decisions you make when writing the code

### Requirements:

The class VecT should have the following declaration:

//=====end of file A2P2.h=========

## //All code, except main() function, in file A2P2.h-----

```
template<typename ET>
class VecT{
public:
   VecT(int = 5);
                                                 // default constructor
   VecT( const VecT & );
                                                 // copy constructor
   VecT( VecT && );
                                                 // move constructor
   ~VecT();
                                                 // destructor
   int getSize() const;
                                                 // accessor for size
   VecT & operator = ( const VecT & ); //copy assignment
   VecT &operator=( VecT && );
                                             // move assignment
   bool operator==( const VecT & ) const;
                                                        // equality operator
   bool operator!=( const VecT &right );
                                                        // not-equal operator
   ET &operator[]( int );
                                       // subscript operator for non-const objects
   ET operator[]( int ) const;
                                           // subscript operator for const objects
private:
   int size;
                     // number of elements in this VecT object
    ET *ptr;
                     //pointer to first element of pointer-based VecT
; // end class VecT
// write the code for the overloaded input/output operators-global non-friend functions here
// all class VecT member functions definitions here
void info();
                     // display on monitor details of the authors of code
```

The following driver program:

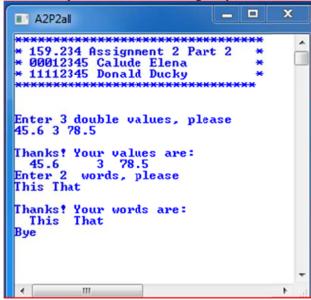
```
- int main() {
    info();
    VecT<double> v1(3);;
    cout<<"\nEnter 3 double values, please\n";
    cin>>v1;

    cout<<"\nThanks! Your values are:\n";
    cout<< v1;

    VecT<string> v2(2);
    cout<<"\nEnter 2 words, please\n";
    cin>>v2;

    cout<<"\nThanks! Your words are:\n";
    cout<<"\nThanks! Your words are:\n";
    cout<<v2;
    cout<<"\nBye\n";
}</pre>
```

produces the following output:



The above example is not testing all member functions.

Make sure that all features (e.g member functions) of your VecT template class work.

**Hand-in**: Submit your completed **A2P2.h** file using the Stream environment.

### **Miscellaneous:**

- 1. Programs that do not compile in the lab, using gcc, get 0 marks.
- 2. Marks will be allocated for: correctness, completeness, use of C++ constructs, programming style, documentation, and clear output.
- 3. Using goto, global variables, C++ tools/constructs that were not presented in lectures 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. When working in teams (at most two students per team), send one solution file per team.
- 5. The assignment will be previewed on Wednesday lecture before it's due.