```
#include <cstdlib>

using namespace std;

#ifndef SLNODE_H
#define SLNODE_H

template<typename T>
class SLNode
{
    public:

    /***** constructor/destructor definitions *****/

        SLNode()
        : nextNode( NULL ), contents( NULL )
        {
            /* empty */
        }

        SLNode( T contents )
        : contents( contents ), nextNode( NULL )
        {
            /* empty */
        }

        ~SLNode()
        {
            nextNode = NULL;
        }

    /***** accessor/mutator function definitions *****/

        void setContents( T &contents )
        {
            this->contents = contents;
        }

        T getContents() const
        {
            return contents;
        }

        void setNextNode( SLNode<T>* nextNode )
        {
            this->nextNode = nextNode;
        }

        SLNode<T>* getNextNode() const
        {
```

```
            return this->nextNode;
        }

        SLNode<T>*& getNextNode()
        {
            return this->nextNode;
        }

    private:

    /***** private variable declarations *****/

        SLNode<T>* nextNode;
        T contents;
};

#endif
```