**CSE225L – Data Structures and Algorithms Lab**

**Lab 11**

**Singly Linked List With Tail**

Add this code to the .h and.cpp file of your code of list that you did in the last class.

|  |
| --- |
| **list.h (addendum)**  #ifndef LIST\_H  #define LIST\_H  template<class T>  class SinglyLinkedListWithTail : public SinglyLinkedList<T>{  protected:  node<T> \*tail;  public:  SinglyLinkedListWithTail();  ~SinglyLinkedListWithTail();  virtual void insertAtStart(T value);  virtual void insertAtEnd(T value);  virtual void deleteEnd();  };  #endif // LIST\_H  **list.cpp (addendum)**  template<class T> SinglyLinkedListWithTail<T>::SinglyLinkedListWithTail()  {  cout<<"In SinglyLinkedListWithTail constructor"<<endl;  tail=NULL;  }  template<class T> SinglyLinkedListWithTail<T>::~SinglyLinkedListWithTail()  {  cout<<"In SinglyLinkedListWithTail destructor"<<endl;  tail=NULL;  }  template<class T> void SinglyLinkedListWithTail<T>::insertAtStart(T value)  {  node<T> \*temp=new node;  temp->data=value;  temp->next=NULL;  if(this->head==NULL) {  this->head=tail=temp;  temp=NULL;  }  else{  temp->next=this->head;  this->head=temp;  }  this->length++;  this->curptr=this->head;  }  template<class T>  void SinglyLinkedListWithTail<T>::insertAtEnd(T value)  {  node<T> \*temp=new node;  temp->data=value;  temp->next=NULL;  if(this->head==NULL) {  this->curptr=this->head=tail=temp;  temp=NULL;  }  else {  tail->next=temp;  tail=temp;  }  (this->length)++;  }  template<class T>  void SinglyLinkedListWithTail<T>::deleteEnd()  {  if(this->isEmpty())throw ListEmpty();  if(this->length==1)  {  this->deleteStart();  return;  }  node<T> \*pre,\*cur=this->head;  while(cur->next!=NULL)  {  pre=cur;  cur=cur->next;  }  tail=pre;  pre->next=NULL;  delete cur;  (this->length)--;  } |

\* In main function, instantiate an object of SinglyLinkedListWithTail<char> class

\* Take some input (char type) values using a loop.

\* Insert those values in start or end position (using insertAtStart() and insertAtEnd() functions) based on user inputs (user will decide whether to insert at start/end).

\* Delete a value (read from user) from the list.