**North South University - Spring 2023**

Course: CSE225L Assessment: Lab 1 / HW 1 / PS 1

Section: 06 NSU ID: 2211424642 Name: Joy Kumar Ghosh

// Task-1 Codes

| #include <iostream>  using namespace std;    const int MAX\_ITEMS = 5;  class FullStack{  // Exception class thrown  // by Push when stack is full.  };    class EmptyStack{  // Exception class thrown  // by Pop and Top when stack is empty.  };      template <class ItemType>  class StackType{  public:  StackType();  bool IsFull();  bool IsEmpty();  void Push(ItemType);  void Pop();  ItemType Top();  private:  int top;  ItemType items[MAX\_ITEMS];  }; |
| --- |
| template <class ItemType>  StackType<ItemType>::StackType(){  top = -1;  }  template <class ItemType>  bool StackType<ItemType>::IsEmpty(){  return (top == -1);  }  template <class ItemType>  bool StackType<ItemType>::IsFull(){  return (top == MAX\_ITEMS-1);  }  template <class ItemType>  void StackType<ItemType>::Push(ItemType newItem){  if(IsFull())  throw FullStack();  top++;  items[top] = newItem;  }  template <class ItemType>  void StackType<ItemType>::Pop(){  if(IsEmpty())  throw EmptyStack();  top--;  }  template <class ItemType>  ItemType StackType<ItemType>::Top(){  if (IsEmpty())  throw EmptyStack();  return items[top];  } |
| //main driver file  int main()  {  StackType<int> integerStack;  int i, item;  char ch;  cout << "Is the stack is empty: ";  if(integerStack.IsEmpty()){  cout << "Stack is Empty" << endl;  }  else{  cout << "Stack is not Empty" << endl;  }  cout <<"\nEnter 4 item for Push: ";  for(i = 0; i < 4; i++){  cin >> item;  integerStack.Push(item);  }  cout << "\nIs the stack is empty: ";  if(integerStack.IsEmpty()){  cout << "Stack is Empty" << endl;  }  else{  cout << "Stack is not Empty" << endl;  }  cout << "\nIs the stack is Full: ";  if(integerStack.IsFull()){  cout << "Stack is Full" << endl;  }  else{  cout << "Stack is not Full" << endl;  }  //printing values  cout << "List: ";  StackType<int> tempStack;  for(int i = 0; !integerStack.IsEmpty(); i++){  tempStack.Push(integerStack.Top());  integerStack.Pop();  }  for(int i = 0; !tempStack.IsEmpty(); i++){  cout << tempStack.Top() << " ";  integerStack.Push(tempStack.Top());  tempStack.Pop();  }  cout << endl << endl;  cout << "Enter another item for push: ";  cin >> item;  integerStack.Push(item);  //printing values  cout << "List: ";  for(int i = 0; !integerStack.IsEmpty(); i++){  tempStack.Push(integerStack.Top());  integerStack.Pop();  }  for(int i = 0; !tempStack.IsEmpty(); i++){  cout << tempStack.Top() << " ";  integerStack.Push(tempStack.Top());  tempStack.Pop();  }  cout << endl << endl;  cout << "\nIs the stack is Full: ";  if(integerStack.IsFull()){  cout << "Stack is Full" << endl;  }  else{  cout << "Stack is not Full" << endl;  }  integerStack.Pop();  integerStack.Pop();  cout << "after pop 2 items, top item is: " << integerStack.Top() << endl << endl;  return 0;  } |
|  |

// Task-2 Codes

|  |
| --- |
|  |
|  |

// Task-3 Codes

|  |
| --- |
|  |
|  |

// Task-4 Codes

|  |
| --- |
|  |
|  |

// Task-5 Codes

|  |
| --- |
|  |
|  |