|  |
| --- |
| /\*  Program 1: hw1  Name: John Carroll  UserID: jcc0044  StudentID#: 902521946  Class: COMP 2710  Date: September 10, 2014  E-Mail: jcc0044@tigermail.auburn.edu  Description: This program will continually prompt for the user name, group name and then  prompt for multiple lines of text messages. (The user name and group name consists of  only one word each.) The text message entry is terminated when in a new line, the user  enters “$$” and the enter key, i.e. a line with only the string “$$”.  \*/  #include <iostream>  #include <string>  using namespace std;  // Function: Main  // Description: Runs the program in the command window.  int main()  {  string inputUserName; // user name that will be entered by user  string userName; // user name placed into cout in the end  string groupName; // group name placed into cout in the end  string inputGroupName; // group name that will be entered by user  string message; // message placed into cout in the end  string inputMessage; // message that the user enters  string inputYesOrNo = "yes"; // yes or no variable that continues or ends the program  string message\_buffer; // string that the user names and messages are stored in for output later  size\_t first\_char;  cout << "===========================================================\n";  cout << "| Welcome to the Auburn Messaging System! |\n";  cout << "===========================================================\n\n\n";  // if user inputs yes for "Any more users?" then the while loop will continue until no is entered.  while (inputYesOrNo == "yes")  {  string newMessage; // placed into the scope for each new entry  cout << "Enter user name> ";  cin >> inputUserName;  newMessage += "|<" + inputUserName;  cout << "Enter group name> ";  cin >> inputGroupName;  newMessage += "::" + inputGroupName + ">|";  cout << "Enter the message> ";  // while loop will continue until user enters $$ and enter key on a single line.  while (inputMessage != "$$")  {  getline(cin, inputMessage);  // If message = $$ then the user is through entering input and the if-statement will not execute  // This condition is important because we do not want to add "$$" to the message\_buffer  if (inputMessage != "$$")  {  // If the message string is empty then it will not add anything to the message\_buffer.  if (inputMessage != "")  {  newMessage += inputMessage + "\n";  }  }  }  message\_buffer = newMessage + message\_buffer;  do{  inputMessage = "";  cout << "\nAny more users? > ";  getline(cin, inputYesOrNo);  cout << "\n";  // transforms the users answer to the question "Any more users?" to lower case.  //std::transform(inputYesOrNo.begin(), inputYesOrNo.end(), inputYesOrNo.begin(), ::tolower);  } while (inputYesOrNo != "yes" && inputYesOrNo != "no");  }  cout << "The current messages are:\n\n";  // This while loop prints out the user names and their messages. As the code executes we remove the part of message\_buffer  // that we no longer need until we reach the end of the string.  while (message\_buffer.size() > 0)  {  first\_char = message\_buffer.find("|<"); // Move message\_buffer to first entry  message\_buffer = message\_buffer.substr(first\_char + 2); // Move message\_buffer  userName = message\_buffer.substr(0, message\_buffer.find("::")); // Get userName  message\_buffer = message\_buffer.substr(message\_buffer.find("::") + 2); // Move message\_buffer  groupName = "(" + message\_buffer.substr(0, message\_buffer.find(">|")) + ")>>\n"; // Get groupName  message\_buffer = message\_buffer.substr(message\_buffer.find(">|") + 2); // Move message\_buffer  // Find either the next beginning of the next entry or everything left in the buffer  message = message\_buffer.substr(0, message\_buffer.find("|<")); // Get message  message\_buffer = message\_buffer.substr(message.size() - 2); // Move message\_buffer  message\_buffer = message\_buffer.substr(message\_buffer.find("\n") + 1); // Move message\_buffer  cout << userName << groupName << message << "\n"; // Pipe to console out  }  cout << ""; // Pipe nothing when done  } |

John Carroll  
Software Construction

Source code Listing and Instructions to Compile using g++

Source code listing

Instructions to compile:

1. Change current directory to correct directory which contains the .cpp using command: “cd”
2. /\* To compile:
   1. type "g++ hw1.cpp -o hw1" in a terminal window to create hw1 executable file; re-executing this will update the hw1 file
   2. type "g++ hw1.cpp" to create and update the a.out with the latest code
   3. type "./a.out" or “./hw1” to run the program \*/