Soccer Club with Polymorphic Inheritance

**Due Time:** 23.59, 16 April 2016 **Earnings:** 9% of your final grade

***NOTE: The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.***

**Purpose:** This is a development of assignment 2 in C++ that uses polymorphism in inheritance. Part of the code (minus the usual headers) is shown on the next page. You **MUST** use this code **without modification (not a single character changed): no code added or removed, no macros, no defines and no statics**. Your task is to implement only the member functions that are declared and not add any new functions. All your project code is in the eleven files Link.h, ClubMember.h, Supporter.h, Player.h, SoccerClub.h, Link.cpp, ClubMember.cpp, Supporter.cpp, Player.cpp, SoccerClub.cpp and ass3.cpp, but you will only submit Link.cpp, ClubMember.cpp, Supporter.cpp, Player.cpp and SoccerClub.cpp.

In this assignment, when the application is running the user can

* Add a new ClubMember at the start of the list with name and subscription and if a Supporter a (unlimited) list of fixture dates and if Player a field position as in ass2. Polymorphism is used through the abstract base class ClubMember from which Player and Supporter are derived. Each link holds a pointer to a ClubMember that is actually the address of a Player or Supporter. Polymorphism ensures that when this ClubMember pointer is used to execute a virtual function, the actual Player or Supporter version gets called.
* Delete the ClubMember at the start of the list
* Print out a particular ClubMember using an overloaded indexing operator
* Print all the ClubMembers starting from the front of the list
* Print all the ClubMembers starting from the end of the list
* Print out a Player for a particular field position

An example of the output of the running application is given at the end. Yours must look identical.

Note the following:

* The SoccerClub::pClubMembers holds the address of the first Link (the head) of the double-linked-list. When a new Club Member (actually a Supporter or a Player) is added a new Link for it is allocated in dynamic (heap) memory and inserted at the head of the list. The new Link has a pointer to the new Club Member (actually a Supporter or a Player) in dynamic memory that has pointers to the first and last names of the player – these are also allocated on the heap in exactly as much memory as they need and no more.
* You must use functions like strlen() and strcpy() or similar etc. from the standard C library to handle strings. You cannot use the C++ string class.
* Input/output is done with cin and cout.
* You must only use new and delete for dynamic memory management. Constructors instantiate objects and destructors release their resources when they are deleted or go out of scope so there are no resource leaks
* When the application terminates it releases **all** dynamically allocated memory (or you lose 30%).

An example of the output of the running application is given at the end. Yours must look identical.

See the Marking Sheet for how you can lose marks, but you will lose 60% if:

1. you change the supplied code in any way at all (not a single character) - no code added or removed, no macros, no defines, no statics and no additional functions,

2. it fails to build in Visual Studio 2013,

3. it crashes in normal operation,

4. it doesn’t work like the example.

There must be no resource leaks (undeleted heap memory) when your application terminates (30% penalty).

Part of the code is shown on the next page. You MUST use this code **without modification.** Your task is to add the implementation of the class member functions in the style of the Submission Standard. Note that each source code file (but not the header files) in general includes both its header and, preceding that, headers of more primitive classes that the compiler needs to know about. Header files do not include other header files except, for example, system header files that may be needed for input/output for inline in inline functions. Each class has its own header file (.h) for its class definition and its own source code file (.cpp) for the bodies of its member functions, defined with scope resolution.

**What to Submit :** Use Blackboard to submit this assignment as a zip file (**not** RAR) containing only the source code files (Link.cpp, ClubMember.cpp, Supporter.cpp, Player.cpp and SoccerClub.cpp). The name of the zipped folder **must** contain your name as a prefix so that I can identify it, for example using my name the file would be tyleraAss3CST8219.zip. It is also vital that you include the Cover Information (as specified in the Submission Standard) as a file headers in your source code files so they can be identified as yours. Before you submit the code, check that it builds and executes in Visual Studio 2013 as you expect - if it doesn’t build for me, for whatever reason, you get a deduction of at least 60%. **Because of Finals it cannot be late**. Don’t send me files as an email attachments – they will get 0.

***Example code: don’t change or add to it (not even a single character), but include the usual headers***

|  |  |
| --- | --- |
| #ifndef CLUBMEMBER\_H  #define CLUBMEMBER\_H  class ClubMember  {  char\* firstName;  char\* lastName;  double subscription;  public:  ClubMember();  ClubMember(char\*,char\*,double);  ClubMember(ClubMember&);  virtual~ClubMember();  virtual void Output()=0;  };  #endif | #ifndef PLAYER\_H  #define PLAYER\_H  class Player:public ClubMember  {  int p;  public:  Player();  Player(char\*, char\*, double, int);  Player(Player&); //copy constructor  ~Player();  bool operator==(Player&);  void Output();  };  #endif |
| #define SUPPORTER\_H  struct Date{  unsigned int day;  unsigned int month;  unsigned int year;  };  class Supporter:public ClubMember  {  Date\* fixtures;  unsigned int numFixtures;  public:  Supporter();  Supporter(char\*, char\*, double, Date\*,unsigned int);  Supporter(Supporter&); //copy constructor  ~Supporter();  void Output();  };  #endif | #ifndef LINK\_H  #define LINK\_H  struct Link  {  ClubMember\* pClubMember;  Link\* pNext;  Link\* pPrev;  Link()  {  pPrev = nullptr;  pNext = nullptr;  }  Link(Link\*, Link\*, ClubMember\*);  ~Link();  };  #endif |
| #define SOCCER\_H  class SoccerClub  {  struct Link\* pClubMembers;  public:  SoccerClub();  ~SoccerClub();  void AddClubMember();  void DeleteClubMember();  void OutputClubMembersForward();  void OutputClubMembersReverse();  void FindPlayer();  ClubMember\* operator[](unsigned int);  friend ostream& operator<<(ostream&, SoccerClub&);  };  #endif |  |
| // ass3.cpp  #include "ClubMember.h"  #include "Supporter.h"  #include "Player.h"  #include "Link.h"  #include "SoccerClub.h"  #include <iostream>  using namespace std;  int main()  {  SoccerClub sc;  bool RUNNING = true;  char response;  int index;  while (RUNNING)  {  cout<<"\nPlease select an option:"<<endl;  cout<<"1. Add a ClubMember"<<endl;  cout<<"2. Delete a ClubMember"<<endl;  cout<<"3. Print out a particular ClubMember"<<endl;  cout<<"4. Print out ClubMembers"<<endl;  cout<<"5. Print out ClubMembers in reverse order"<<endl;  cout<<"6. Find a Player for a particular position"<<endl;  cout<<"q. Quit"<<endl;  cout<<"CHOICE: ";  cin>>response;  switch (response)  {  case '1':sc.AddClubMember(); break;  case '2':sc.DeleteClubMember(); break;  case '3':  cout << "please enter the player index: ";  cin >> index;  sc[index] != nullptr ? sc[index]->Output() : cout << "Invalid Index"<<endl;  break;  case '4':cout<<sc; break;  case '5':sc.OutputClubMembersReverse(); break;  case '6':sc.FindPlayer(); break;  case 'q': return 0;  default:cout<<"Please enter a valid option\n";  }  cout<<"\n";  }  } | |

***Example Output (yours must work identically)***

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE: 1**

**ADDING A CLUB MEMBER**

**Please enter the Club Member first name: John**

**Please enter the Club Member last name: Smith Jones**

**Please enter the Club Member subscription: 100.00**

**Supporter (S) or Player (P)?**

**S**

**Supporter**

**Please enter the number of fixtures**

**3**

**enter the day of fixture #0 02**

**enter the month of fixture #0 02**

**enter the year of fixture #0 2017**

**enter the day of fixture #1 12**

**enter the month of fixture #1 03**

**enter the year of fixture #1 2017**

**enter the day of fixture #2 5**

**enter the month of fixture #2 07**

**enter the year of fixture #2 2017**

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE: 1**

**ADDING A CLUB MEMBER**

**Please enter the Club Member first name: Mary**

**Please enter the Club Member last name: Johnson**

**Please enter the Club Member subscription: 200.00**

**Supporter (S) or Player (P)?**

**P**

**Player**

**Please enter the Player position:**

**Goalkeeper = 0,**

**left-back = 1,**

**right-back = 2,**

**centre-back = 3,**

**left-midfield = 4,**

**right-midfield = 5,**

**centre-midfield = 6,**

**centre-forward = 7:**

**3**

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE: 4**

**FORWARD PRINTING CLUB MEMBERS**

**Club Member 0**

**Player**

**Mary Johnson**

**Subscription = 200**

**Position = centre-back (3)**

**Club Member 1**

**Supporter**

**John Smith Jones**

**Subscription = 100**

**fixtures [0] = 2/2/2017**

**fixtures [1] = 12/3/2017**

**fixtures [2] = 5/7/2017**

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE: 6**

**Please enter the position you are searching for: 3**

**position found**

**Player**

**Mary Johnson**

**Subscription = 200**

**Position = centre-back (3)**

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE: 3**

**please enter the player index: 1**

**Supporter**

**John Smith Jones**

**Subscription = 100**

**fixtures [0] = 2/2/2017**

**fixtures [1] = 12/3/2017**

**fixtures [2] = 5/7/2017**

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE: 2**

**DELETE A CLUB MEMBER**

**Club Member deleted**

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE: 3**

**please enter the player index: 1**

**Invalid Index**

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE: 3**

**please enter the player index: 0**

**Supporter**

**John Smith Jones**

**Subscription = 100**

**fixtures [0] = 2/2/2017**

**fixtures [1] = 12/3/2017**

**fixtures [2] = 5/7/2017**

**Please select an option:**

**1. Add a ClubMember**

**2. Delete a ClubMember**

**3. Print out a particular ClubMember**

**4. Print out ClubMembers**

**5. Print out ClubMembers in reverse order**

**6. Find a Player for a particular position**

**q. Quit**

**CHOICE:**