Soccer Club with Overloaded Operators

**Due Time:** 23.59, 19 Mar 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 1 in C++ that uses a copy constructor for pass-by-value and overloaded operators. 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 seven files Link.h, Player.h, SoccerClub.h, Link.cpp, Player.cpp, SoccerClub.cpp and ass2.cpp, but you will only submit Link.cpp, Player.cpp and SoccerClub.cpp.

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

* Add a new Player at the start of the list with name, subscription and field position
* Delete the player at the start of the list
* Print out a particular player using an overloaded indexing operator
* Print all the players starting from the front of the list
* Print all the players 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::pPlayers holds the address of the first Link (the head) of the double-linked-list. When a new 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 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, 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 tyleraAss2CST8219.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%. There is a late penalty of 25% per day. 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***

|  |  |  |
| --- | --- | --- |
| // In Link.h  #ifndef LINK\_H  #define LINK\_H  struct Link  {  Player\* pPlayer;  Link\* pNext;  Link\* pPrev;  Link()  {  pPrev = nullptr;  pNext = nullptr;  }  Link(Link\*,Link\*,Player\*);  ~Link();  };  #endif | // In Player.h  #ifndef PLAYER\_H  #define PLAYER\_H  class Player  {  private:  char\* firstName;  char\* lastName;  double subscription;  int p;  public:  Player();  Player(char\*,char\*,double,int);  Player(Player&);  ~Player();  void Print();  bool operator==(Player&);  };  #endif | // In SoccerClub.h  #ifndef SOCCER\_H  #define SOCCER\_H  class SoccerClub  {  Link\* pPlayers;  public:  SoccerClub();  ~SoccerClub();  void AddPlayer();  void DeletePlayer();  void PrintPlayersForward();  void PrintPlayersReverse();  void FindPlayer();  Player\* operator[](unsigned int);  friend ostream& operator<<(ostream&, SoccerClub&);  };  #endif |
| // In ass2.cpp  void Print(Player p)  {  p.Print();  }  int main()  {  SoccerClub sc;  bool RUNNING = true;  char response;  int index;  while (RUNNING)  {  cout<<"\nPlease select an option:"<<endl;  cout<<"1. Add a Player"<<endl;  cout<<"2. Delete a Player"<<endl;  cout<<"3. Print out a particular Player"<<endl;  cout<<"4. Print out Players"<<endl;  cout<<"5. Print out Players 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.AddPlayer(); break;  case '2':sc.DeletePlayer(); break;  case '3':  cout << "please enter the player index: ";  cin >> index;  sc[index] != nullptr ? Print(\*(sc[index])) : cout << "Invalid Index"<<endl;  break;  case '4':cout<<sc; break;  case '5':sc.PrintPlayersReverse(); 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 Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 1***

***ADDING AN PLAYER***

***Please enter the Player first name: Bill John***

***Please enter the Player last name: Smith Jones***

***Please enter the Player subscription: 100.00***

***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:***

***2***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 1***

***ADDING AN PLAYER***

***Please enter the Player first name: Mary Jo***

***Please enter the Player last name: Bell***

***Please enter the Player subscription: 200.00***

***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:***

***7***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 3***

***please enter the player index: 0***

***Last Name = Bell***

***First Name = Mary Jo***

***Subscription = 200***

***Position = centre-forward (7)***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 6***

***Please enter the position you are searching for: 2***

***position found***

***Last Name = Smith Jones***

***First Name = Bill John***

***Subscription = 100***

***Position = right-back (2)***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 6***

***Please enter the position you are searching for: 9***

***position not found***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 3***

***please enter the player index: 3***

***Invalid Index***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 4***

***FORWARD PRINTING PLAYERS***

***Player 0***

***Last Name = Bell***

***First Name = Mary Jo***

***Subscription = 200***

***Position = centre-forward (7)***

***Player 1***

***Last Name = Smith Jones***

***First Name = Bill John***

***Subscription = 100***

***Position = right-back (2)***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 2***

***DELETE A PLAYER***

***Player deleted***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE: 4***

***FORWARD PRINTING PLAYERS***

***Player 0***

***Last Name = Smith Jones***

***First Name = Bill John***

***Subscription = 100***

***Position = right-back (2)***

***Please select an option:***

***1. Add a Player***

***2. Delete a Player***

***3. Print out a particular Player***

***4. Print out Players***

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

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

***q. Quit***

***CHOICE:***