

CSC 420/520 Spring 2013 Lab 1

This lab assignment is due Tuesday, 01/22/2013 at **6pm**. See the syllabus for late penalties.

You should turn it in by blackboard. Each task should be in a separate unzipped folder named **YourLastName_YourFirstInitial_taskN**, where N is the task number inside a zipped file named: **YourLastName_YourFirstInitial.zip**.

You may use code you have written previously and you may ask other members of the class questions about the assignment. However, you must do your own assignment; you may not use anyone else's code. **You must write your own code.** Also, remember any task that doesn't compile will receive 0 points.

Task 1 - 40 points

Implement a linked list.

Write a C++ class (or set of classes) that implements a linked list. The data contained in the nodes are integers. The maximum number of nodes in the list is 100. Implement at least the following functions:

- Constructor and destructor;
- **void insert (int number);** - inserts an element with the value number at the beginning of the list;
- **int isEmpty();** - returns 1 if the list is empty, 0 otherwise;
- **int numOfElements();** - returns the number of elements in the list.

You do not need to include any other classes for full credit, but you can if you choose.

Clean the solution of the project and put the entire project including source code in a folder named **YourLastName_YourFirstInitial_task1**.

Task 2 – 45 points

Since it's fall we need to write a program to help our favorite team keep track of its players performance. The file that will print out the player's stats is done. You are given a class for a player; however, a football team is composed of offensive and defensive players. So you need to build on the player class. It is rare that a player is both, so we will assume our team has exclusively offensive and defensive players. All players have a name, number and they all play for some number of minutes. However, a defensive player is measured by the number of tackles they get, while an offensive player is measured by the number of yards they get. For this assignment you need to create 2 classes that inherit from a class, player that has already been written. This class makes no calculations; it just holds data on each player. Your class will be used by the main function in main.cpp to print the team's stats, so it must follow the guidelines below. You may not change the files given in any way.

Download the Football project and unzip it. It will not compile because it is missing some classes. **The main.cpp, player.h and player.cpp files must not be changed in any way.** Ten points will be deducted for each modified file.

The project needs 4 files added: defense.h, defense.cpp, offense.h, offense.cpp. There are also a few questions that must be answered.

The files should be formatted similar to the player class files. The classes need to have the following:

defense class variables:

```
int tackles;
```

defense class methods:

```
Defense(string name);  
setMinutesPlayed(int minutes);  
setTackles(int tackles);  
printStats() const;
```

offense class variables:

```
int yards;
```

offense class methods:

```
Offense(string name);  
setMinutesPlayed(int minutes);  
setYards(int yards);  
printStats() const;
```

*Note: You must decide on the return types, visibility and any additional modifiers for these methods.

Rules:

Your classes must inherit from the base class player.

You should not recode anything. If the base class already does something, don't put it in your derived class. Use what is already there.

You should follow the coding conventions of the program given.

Do not include .cpp files

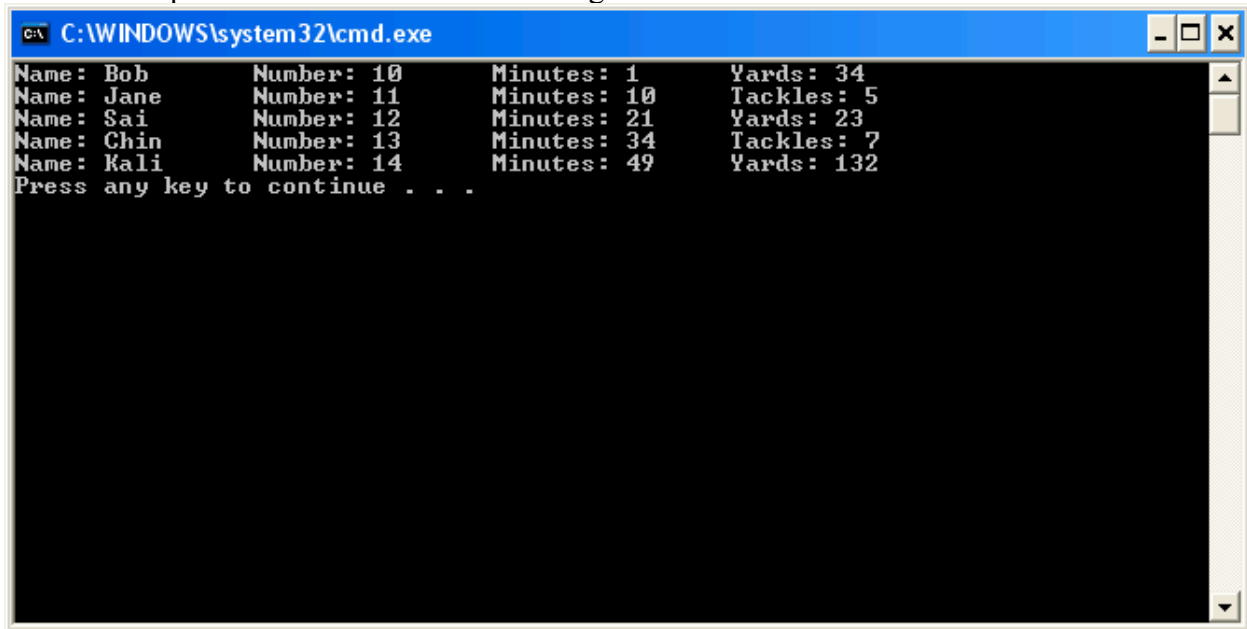
All files should have a brief description at the beginning that includes the title of the file and your name. The code should have comments describing each method and large section. See the provided code for examples.

All 4 of the methods listed above must be implemented. You may add methods, but it is not necessary.

This is just a suggestion, but the solution averages exactly one line of code inside each method, so if a method has 2 lines of code another should have 0 lines of code. There is no trick here either with crazy lines that call multiple functions or anything else. Basically, if you are writing many lines of code per method, you need to review inheritance or polymorphism.

All classes should include all needed libraries.

The final output should look like the following:



```
C:\WINDOWS\system32\cmd.exe
Name: Bob      Number: 10     Minutes: 1     Yards: 34
Name: Jane     Number: 11     Minutes: 10    Tackles: 5
Name: Sai      Number: 12     Minutes: 21    Yards: 23
Name: Chin     Number: 13     Minutes: 34    Tackles: 7
Name: Kali     Number: 14     Minutes: 49    Yards: 132
Press any key to continue . . .
```

Questions:

- 1) On lines 29 to 33 of main.cpp, why is there an '&' before the player variables?
- 2) For lines 45 to 49 of main.cpp, why not use a loop?
- 3) On lines 10, 11 and 52 of player.h, what do the #ifndef, #define and #endif preprocessor commands do?
- 4) On line 20 of player.h, the string name is private, what did you have to do because of this?
- 5) On lines 24 of player.h, why does minutes played have to be protected, not private?

- 6) On line 43 of player.h, why is the function assigned zero?
- 7) On line 49 of player.h, what does the word 'const' do?
- 8) What method should be entirely in the base class? Why?
- 9) Which class is an abstract class? Why?
- 10) Name one thing you learned or had forgotten and remembered in this task.

From the menu "Build-> Clean Solution" of Visual Studio remove the binary files from the project.

Answer the above 10 questions in a text file and the cleaned solution of the project with the source code and put it all in a folder named **YourLastName_YourFirstInitial_task2**.

Task 3 – 10 points

Open ANote project, from the zip folder you downloaded and build the project. There is one compilation error. Fix it.

From the menu "Build-> Clean Solution" of Visual Studio remove the binary files from the project.

Clean the solution of the project and put the entire project including source code in a folder named **YourLastName_YourFirstInitial_task3**.

To turn in your assignment, place all 3 folders into a zipped folder called, **YourLastName_YourFirstInitial.zip**. **Only the top folder should be zipped**. Then upload it to blackboard, download it yourself to make sure it was uploaded correctly. If you file is not upload correctly, you will be subject to the late policy.

Task 4 – 5 points

Read chapter 1 and chapter 2 in the text book and answer the questions at the end of each chapter. Chapter 1 is titled "History of Software Engineering" and Chapter 2 is titled "Software Life Span Modules". You will receive the credit for this activity by turning in your answers to the questions at the end of each chapter. These chapters are light reading and provide a foundation for the rest of the course.