Crib Sheet 1

Wednesday, February 1, 2023 9:42 PM

Player.h

```
# include <iostream>
// Purpose: header class for player class
class Player {
        public: // public data
        Player (std::string player, std::string team);

        // Accesors
        // const to not change variable
        Std:: string getplayer () const;
        std:: string get team () const;
        int get goals () const;
        int getassists() const;
        int get penalities() const;

         //modifiers
        Void add goals (int i = 1);
        void add assists (int i=1);
        Void addpenalities ( int i=1);

        // Representation
        private:
        Std:: string player;
        std:: string team;
        int goals;
        int assists;
        int penalities;
}; // never forget semicolon in header
bool stronger (const Player &player1, const Player &player2);
```

Player.cpp

```
# include <iostream>
# include "player.h" // includes header
/* if Player class had a default constructor:
 Player:: Player () {
      player = "None";
      Team = "None";
      goals = 0;
      assists = 0;
       penalities = 0;
*/
// Purpose : Implementation of the player class
Player:: Player (std::string aplayer , std::string ateam) {
     player = aplayer;
     team = ateam;
     goals = 0;
     assists = 0;
     penalities = 0;
}
Std:: string Player:: get player() const {
      return player;
}
Std:: string Player:: get team() const {
      return team;
}
int Player:: get goals() const {
      return goals;
}
int Player:: get assists () const {
      return assists;
}
int Player:: get penalities () const {
      return penalities;
}
Void Player:: add goals (int i) {
      // i is already initialized.
      goals += i;
}

void Player:: add assists (int i) {
    assists += i;
}
Void Player:: add penalities (int i) {
     penalities += i;
}
/* even though stronger function is outside of
   header, it is included here because it is in the
    file. It isn't a player function though
*/
```

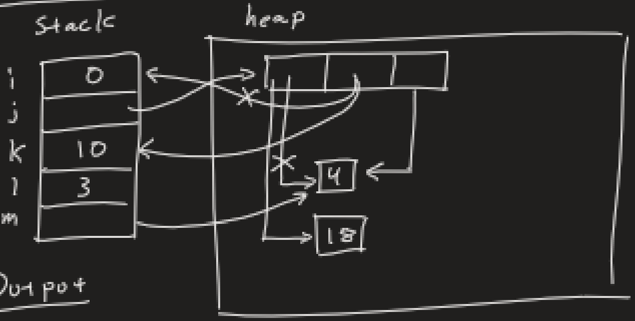
```
//Compares 2 players by goals & assists, least penalities, then
// alphabetically
bool stronger (const Player& player1, const Player& player2) {
     int player1_stats = player1.getgoals() + player1.getassists();
     int player2_stats = player2.get goals() + player2.getassists();
     if (player1_stats != player2_stats) {
         return (player1_stats > player2_stats);
     } else if (player1.getpenalities() != player2.get penalities()) {
         return (player1.get penalities < player 2.get penalities());
     } else {
         return (player.get player() < player 2.get player());
```

Memory Diagram

Code:

```
int i,  **j , k, l, *m;
i=0;
j= new int *[3];
j[0] = new int;
j[1] = &i;
m= *(j+1);
j[1]= &k
k=10;
*(j[0])= 5;
j[2]=j[0];
*(j[0]) = 18;
*m=4;
l=3
```

Diagram



Output

```
i:  0
j[0]: 18
j[1]: 10
j[2]: 4
k: 10
l:  3
m: 4
```

Strings

include <ctypes>

```
string.isalpha()   string.isdigit()
//returnsa bool whether string is letter/digit
String.islower()   string.isupper()
 // returns bool whether string is lower/upper
String.tolower()   string.toupper()
```

Vectors

```
#include <vector>
std:: vector <int> a(10); // vector size 10 of ints
a.back(4); //a[-1] = 4
a. pop_back(); //vector size is now 9
 a. front(); // a[0]
a[20]; // memory not given to you, seg fault
```

Consts & &

```
 & //reference.  use in function so time is less
 ...... const() // won't modify variables
                // no affect on run time.
 const .....   // passes a reference and no
               // changes.
```

File reading

```
# include <fstream>
 std::ifstream in_str("file.txt"); //reads files
 std::ofstream out_str("file.txt"); // writes files
 if (!in_str.good()){ // checking if text file exist
     std::cerr << "Can't Open" << std::endl;
     exit(1);

in_str >>x;
Out_str << y << std::endl;
in_str.close();
out_str.close();
```

```
Sizeof(x) // tells you how many bytes it holds.
          // divide by variable type to get number of elements
```

Errors

std has no member → missing an include, not found in library

redefinition of "class Class" → add guards to class.h file

"reference to local variable" → remove & to variable

"control reaches end of non-void function" → force return a
State Menu

Uninitialized Read → didn't declare "new"

unaddressable Access → accessing past array size

Invalid heap argument & LEAK → didn't use delete