Criterion C: Development

The Mainclass has six case statements performing functions of (add, edit, printout, delete writetofile and read from file.

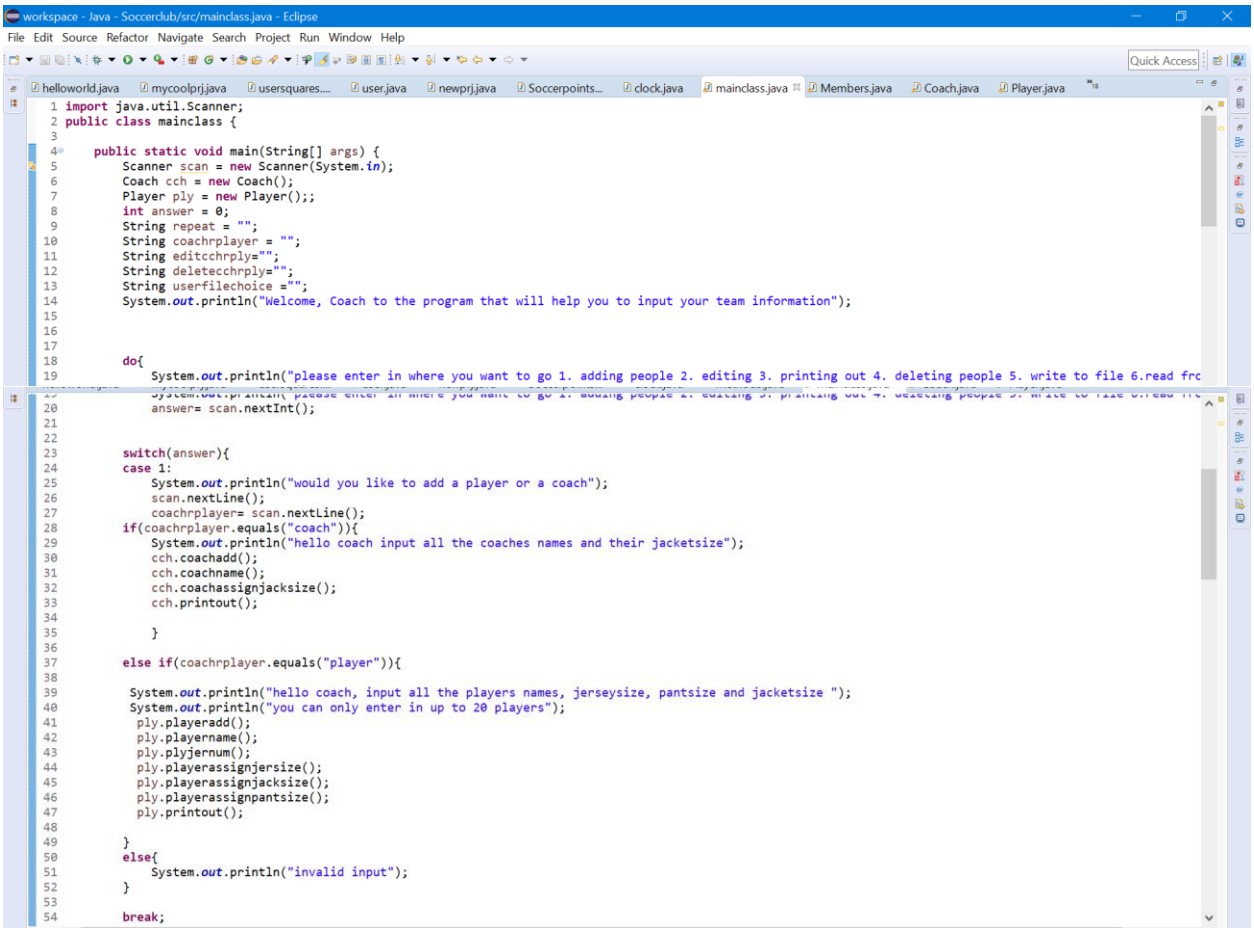
The program begins with a do while statement. The do while has been chosen so that the program will always go through the case statements once before evaluating repeat. This helps the user to decide at the end of the program if they will repeat or not. As, can be seen in the screenshot below, the repeat evaluates at the end of the program.

```
16
17
18        do{
19            System.out.println("please enter in where you want to go 1. adding people 2. editing 3. printing out 4. deleting people 5. write to file 6.read frc
20            answer= scan.nextInt();
21
```

```
149
150           System.out.println(" if you want to repeat 'yes' ");
151           repeat = scan.nextLine();
152       }while(repeat.equals("yes"));
153
```

Case 1: Performing the add function

Mainclass:



The user has a choice to either choose player or coach. The program then executes the respective if and else if statements.

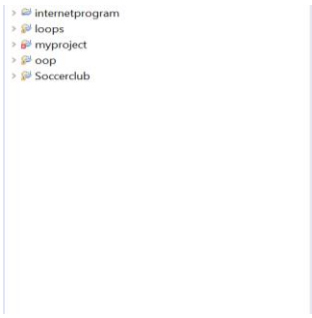
If coach is chosen, then the methods are called from the Coach class

Coach class:

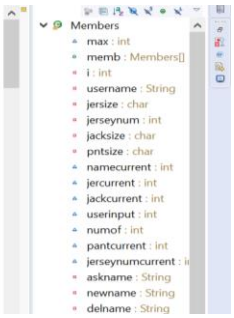
```
1*import java.io.FileInputStream;
6
7  public class Coach extends Members {
8      private String askname;
9      private String newname;
10     private char newjacksize;
11     private String delname;
12     private String username;
13     private int cchnamecurrent;
14     private boolean cchedit=false;
15     private boolean cchdel = false;
16     Scanner myscan = new Scanner(System.in);
17
18     public void coachadd(){
19      super.addmember();
20
21
22         }
23     public void coachassignjacksize(){
24         super.assignjacksize();
25     }
26     public void coachname()
27     {
28         for(int i =0; i<numof ; i++){
29             System.out.println("enter in the name of coach " + (cchnamecurrent+1));
30             username = myscan.nextLine();
31             memb[cchnamecurrent].name(username);
32             cchnamecurrent = cchnamecurrent+1;
33         }
34     }
35     public void printout()
36     {
37         for( int i=0; i<userinput; i++){
38             System.out.println("The name of Coach " + (i+1) + '\t' + memb[i].getusername());
39             System.out.println("The jacket size of Coach " + (i+1) + '\t' + memb[i].getjacksize());
40
41             }
42     }
```

The main example of inheritance is the memb object array(seen below) which is used for both the Coach and Player when they are created in the mainclass. This was used so that a separate object array would not have to be created for Player and Coach. The coachadd() and coachassignjacksize() are called from the Members class which use inheritance. The coachname() method uses polymorphism, mutating the method assignname() in Members. Some methods were mutated because they needed to be specific to either coach or player (such as the printout method). This shows how OOP is used in the program.

Members class:




```
1*import java.util.Scanner;
6  public class Members extends Persons   {
7
8      int max =21;
9      public Members[] memb = new Members[max];
10     private int i=0;
11     private String username;
12     private char jersize;
13     private int jerseynum;
14     private char jacksize;
15     private char pntsize;
16     int namecurrent=0;
17     int jercurrent=0;
18     int jackcurrent=0;
19     int userinput=0;
20     int numof=0;
21     int pantcurrent=0;
22     int jerseynumcurrent;
23     private String askname;
24     private String newname;
25     private String delname;
26     private char newjersize;
27     private char newjacksize;
28     Scanner myscan = new Scanner(System.in);
```



```
public void addmember(){
    System.out.println("how many people would you like to add");
    numof = myscan.nextInt();
    myscan.nextLine();
    if ((numof + userinput) < memb.length){
        System.out.println(numof + userinput);
        for(int i = 0 ; i<numof; i++)
        {
            memb[userinput] = new Members();
            userinput = userinput +1;
                }
        }
    else
    {
        System.out.println(numof+" new members will exceed the number of legit members");
    }

}
public void assignname(){

        for(i =0; i<numof ; i++){
        System.out.println("enter in the name of person " + (namecurrent+1));
        username = myscan.nextLine();
        memb[namecurrent].name(username);
        namecurrent = namecurrent+1;

        }

    }
    public void assignjersize(){
        for(i =0; i<numof; i++){
        System.out.println("type 's' for small, 'm' for medium and 'l' for large");
        System.out.println("enter the jersize for " + memb[jercurrent].getusername());
        jersize = myscan.next().charAt(0);
        memb[jercurrent].jersize(jersize);
        jercurrent = jercurrent+1;
```

Updates Available
Updates are available for your software.
Click to review and install updates.

Case1: Sample output with two coaches

```
Problems  Javadoc  Declaration  Console
mainclass [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Mar 18, 2018, 10:55:29 AM)
Welcome, Coach to the program that will help you to input your team information
please enter in where you want to go 1. adding people 2. editing 3. printing out 4. deleting people 5. write to file 6.read from file
1
would you like to add a player or a coach
coach
hello coach input all the coaches names and their jacketsize
how many people would you like to add
2
2
enter in the name of coach 1
Tom
enter in the name of coach 2
Tris
type 's' for small, 'm' for medium and 'l' for large
enter in the jacket size for Tom
m
type 's' for small, 'm' for medium and 'l' for large
enter in the jacket size for Tris
l
The name of Coach 1     Tom
The jacket size of Coach 1      m
The name of Coach 2     Tris
The jacket size of Coach 2      l
 if you want to repeat 'yes'
```

If player is chosen, then the methods are called from the Player class.

Player class:

```
private int newjernum;
private String username;
private int plynamecurrent;
Scanner myscan = new Scanner(System.in);

public void playeradd(){

     super.addmember();


    }
    public void playerassignjacksize(){
        super.assignjacksize();
    }
    public void playername()
    {
        for(int i =0; i<numof ; i++){
            System.out.println("enter in the name of player " + (plynamecurrent+1));
            username = myscan.nextLine();
            memb[plynamecurrent].name(username);
            plynamecurrent = plynamecurrent+1;
        }
    }
    public void printout()
    {
        for(int i = 0; i<userinput; i++){
            System.out.println("The name of player " + (i+1) + '\t' + memb[i].getusername());
            System.out.println("the jerseynumber for player " + (i+1) + '\t' + "is " + memb[i].getjerseynum() );
            System.out.println("the jersey size of player " + (i+1) + '\t' +  memb[i].getjersize());
            System.out.println("The jacket size of player " + (i+1) + '\t' + memb[i].getjacksize());
            System.out.println(" the shortsize of player" + (i+1) + '\t' + memb[i].getpntsize());

        }

    }
    public void playerassignjersize()
    {
        super.assignjersize();

    }
    public void plyjernum(){
        super.jerseynummemb();
    }
    public void playerassignpantsize(){
        super.assignpantsize();
    }
```

Updates Available
Updates are available for your software.
Click to review and install updates.

Some methods are inherited from Members class, while some are modified in Player class. These were modified because they needed to be specific to Player(ex. printout, which prints other attributes like jerseynum or jersize different from Coach). Each method also has its own counter variable. This was chosen so that the counter for that method would only be used for the methods purpose(ex. jercurrent is for jersey size only).

Members class:

```
62      public void assignjersize(){
63          for(i =0; i<numof; i++){
64          System.out.println("type 's' for small, 'm' for medium and 'l' for large");
65          System.out.println("enter the jersize for " + memb[jercurrent].getusername());
66          jersize = myscan.next().charAt(0);
67          memb[jercurrent].jersize(jersize);
68          jercurrent = jercurrent+1;
69          }
70
71
72      }
73      public void jerseynummemb(){
74          for(i =0; i<numof; i++){
75              System.out.println("enter the jerseynumber for " + memb[jerseynumcurrent].getusername());
76              jerseynum = myscan.nextInt();
77              memb[jerseynumcurrent].jerseynumber(jerseynum);
78              jerseynumcurrent= jerseynumcurrent+1;
79          }
80      }
81      public void assignjacksize(){
82          for(i=0; i<numof; i++){
83              System.out.println("type 's' for small, 'm' for medium and 'l' for large");
84              System.out.println("enter in the jacket size for " + memb[jackcurrent].getusername());
85              jacksize = myscan.next().charAt(0);
86              memb[jackcurrent].jacksize(jacksize);
87              jackcurrent = jackcurrent +1;
88          }
89      }
90      public void assignpantsize(){
91          for(i = 0; i<numof; i++){
92              System.out.println("type 's' for small, 'm' for medium and 'l' for large");
93              System.out.println("enter in the shortsize for " + memb[pantcurrent].getusername());
94              pntsize = myscan.next().charAt(0);
95              memb[pantcurrent].pantsize(pntsize);
96              pantcurrent = pantcurrent +1;
```

Updates Available
Updates are available for your software.
Click to review and install updates.

Case 1: sample output with two players

```
mainclass [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Mar 18, 2018, 11:33:54 AM)
Welcome, Coach to the program that will help you to input your team information
please enter in where you want to go 1. adding people 2. editing 3. printing out 4. deleting people 5. write to file 6.read from file
1
would you like to add a player or a coach
player
hello coach, input all the players names, jerseysize, pantsize and jacketsize
you can only enter in up to 20 players
how many people would you like to add
2
2
enter in the name of player 1
Cameron
enter in the name of player 2
Jon
enter the jerseynumber for Cameron
12
enter the jerseynumber for Jon
10
type 's' for small, 'm' for medium and 'l' for large
enter the jersize for Cameron
s
type 's' for small, 'm' for medium and 'l' for large
enter the jersize for Jon
m
type 's' for small, 'm' for medium and 'l' for large
enter in the jacket size for Cameron
m
type 's' for small, 'm' for medium and 'l' for large
enter in the jacket size for Jon
m
type 's' for small, 'm' for medium and 'l' for large
enter in the shortsize for Cameron
m
type 's' for small, 'm' for medium and 'l' for large
enter in the shortsize for Jon
m
The name of player 1    Cameron
the jerseynumber for player 1   is 12
the jersey size of player 1     s
The jacket size of player 1     m
 the shortsize of player1       m
The name of player 2    Jon
the jerseynumber for player 2   is 10
the jersey size of player 2     m
The jacket size of player 2     m
 the shortsize of player2       m
 if you want to repeat 'yes'
```

Updates Available
Updates are available for your software.
Click to review and install updates.
Updates Available
Updates are available for your software.

Case 2: Performing the edit function

Mainclass:

```
55          case 2:
56              System.out.println("you will be editing the database");
57              System.out.println("who do you want to edit, a coach or player");
58              scan.nextLine();
59              editcchrply = scan.nextLine();
60          if(editcchrply.equals("coach") & (cch.userinput>0)){
61              cch.printout();
62              cch.coachedit();
63          }
64          else if(editcchrply.equals("player") & (ply.userinput>0)) {
65              ply.printout();
66              ply.playeredit();
67          }
68          else{
69              System.out.println("you either do not have a coach or player entered, please add them");
70          }
71          break;
```

The program has been designed here so that both conditions of editcchrply and userinput >0 have to be true in order for the program to continue with the if statements. The check for userinput is for the case that no coach or players have been entered. When there are no coach or players then the else statement will execute and tell the user to add people.

User chooses coach:

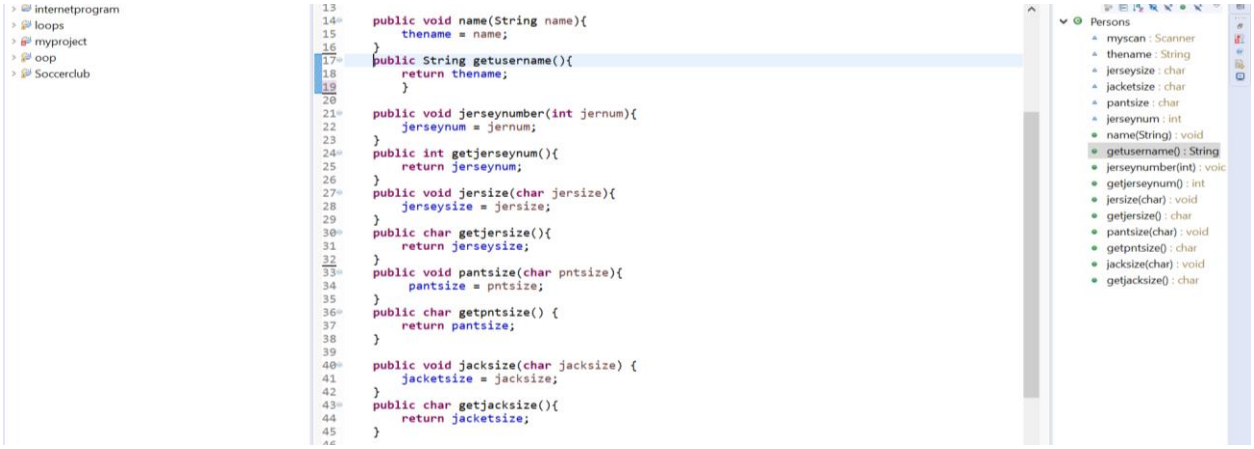
Coach class

```
44      public void coachedit(){
45          System.out.println("which person do you want to edit");
46          askname = myscan.nextLine();
47          for( int i = 0; i<userinput; i++){
48          if(askname.equalsIgnoreCase( memb[i].getusername())){
49              System.out.println("found name");
50              System.out.println("what would you like to change it to");
51              newname = myscan.nextLine();
52               memb[i].name(newname);
53          System.out.println(memb[i].getusername());
54          System.out.println("what would you like to change the jacketsize to");
55          newjacksize = myscan.next().charAt(0);
56          memb[i].jacksize(newjacksize);
57          cchedit = true;
58          }
59
60
61          }
62          if(cchedit){
63              System.out.println("editing has finished");
64          }
65          else{
66              System.out.println("invalid input");
67          }
```

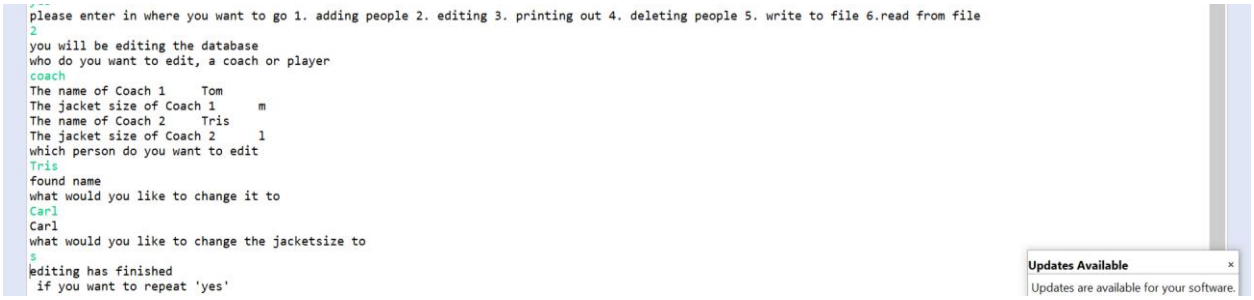
Updates Available
Updates are available for your software.
Click to review and install updates.

The mutator methods called from the Persons class(seen below) makes it easier to change all the attributes of the object just by assignment. Which is why the use of accessor and mutator methods were adopted for the program. It makes it simpler to execute all the methods. The accessor gets the attribute from the object and the mutator assign a new value for the attribute(ex.jacksize). In Addition, boolean cchedit was adopted for the purpose of checking if the coachedit() runs successfully(same with Player class). The Boolean checks the execution of the method then, outputs a message based on the value of the Boolean outside the loop.

Persons Class

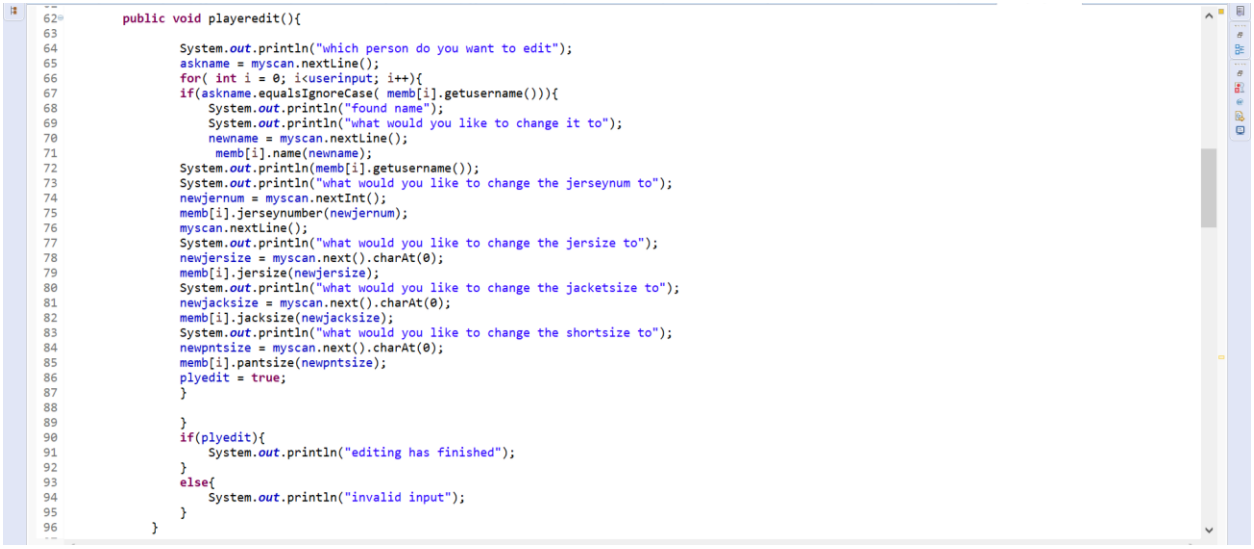


Case 2: Sample output with coach edit



User chooses player

Player Class



Playeredit() functions in the same way as coachedit() but, the difference being the editing of the extra attributes of the Player object. The Boolean plyedit functions the same as in Coach class.

Case 2: Sample output with player edit

```
please enter in where you want to go 1. adding people 2. editing 3. printing out 4. deleting people 5. write to file 6.read from file
2
you will be editing the database
who do you want to edit, a coach or player
player
The name of player 1    Cameron
the jerseynumber for player 1   is 12
the jersey size of player 1     s
The jacket size of player 1     m
 the shortsize of player1       m
The name of player 2    Jon
the jerseynumber for player 2   is 10
the jersey size of player 2     m
The jacket size of player 2     m
 the shortsize of player2       m
which person do you want to edit
Jon
found name
what would you like to change it to
Barron
Barron
what would you like to change the jerseynum to
10
what would you like to change the jersize to
s
what would you like to change the jacketsize to
s
what would you like to change the shortsize to
s
editing has finished
 if you want to repeat 'yes'
```

Case 3: Performing printout

Mainclass

The printout's purpose is to allow the user to see the present players and coaches. The benefit of the if statements is that the program will print by determining the value of userinput. This allows the program to avoid any errors and unwanted exceptions. The printout() uses accessor methods for each attribute of the object thus, making it simpler to use.

Coach Class

```
public void printout()
{
    for( int i=0; i<userinput; i++){
        System.out.println("The name of Coach " + (i+1) + '\t' + memb[i].getusername());
        System.out.println("The jacket size of Coach " + (i+1) + '\t' + memb[i].getjacksize());

        }
}
```

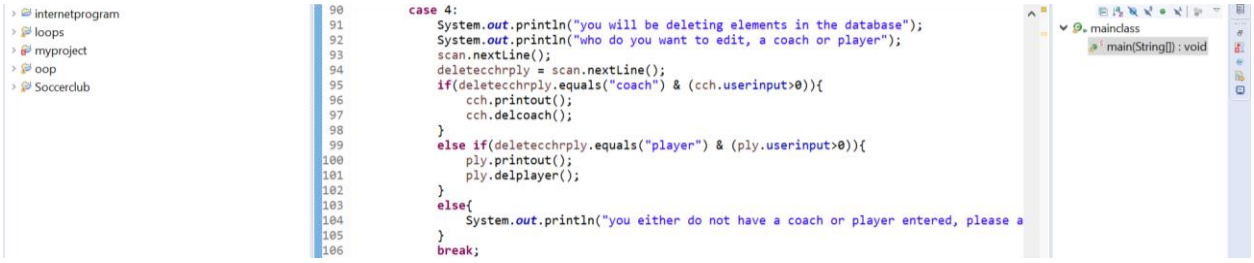
Player class

```
    public void printout()
    {
        for(int i = 0; i<userinput; i++){
            System.out.println("The name of player " + (i+1) + '\t' + memb[i].getusername());
            System.out.println("the jerseynumber for player " + (i+1) + '\t' + "is " + memb[i].getjerseynum() );
            System.out.println("the jersey size of player " + (i+1) + '\t' +  memb[i].getjersize());
            System.out.println("The jacket size of player " + (i+1) + '\t' + memb[i].getjacksize());
            System.out.println(" the shortsize of player" + (i+1) + '\t' + memb[i].getpntsize());

        }

    }
```

The printout() Outputs have already been seen in other case statements

Case 4: Performing Delete

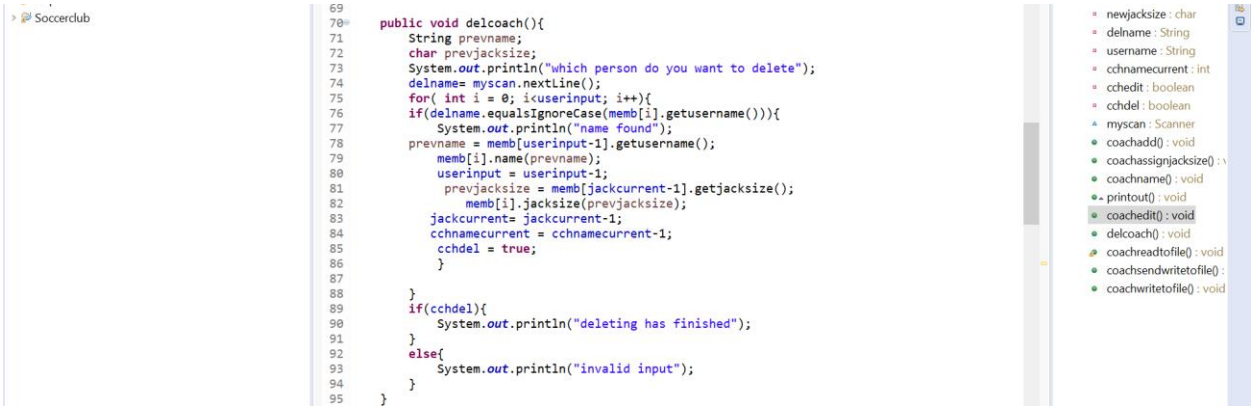
Mainclass



The if statements are designed in the same way as the edit case (i.e. evaluation of both deletecchrply and userinput).

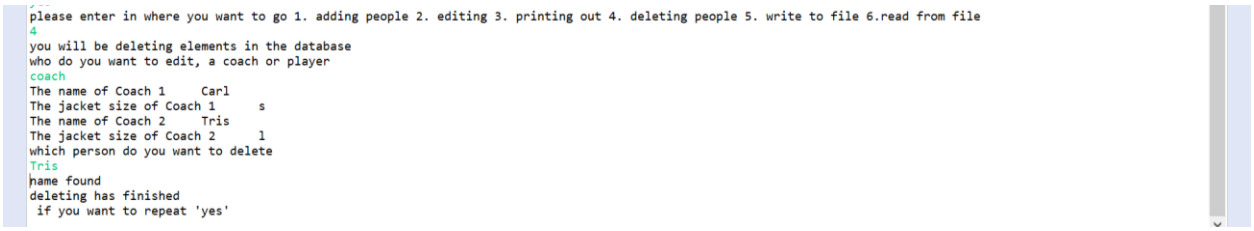
User chooses coach:

Coach Class



The same aspect of checking cchdel is used here as in the edit case. This checks if the delete has finished. In this case, accessor methods store the value of the last “member’s” attribute and place it in the current position in the loop using mutator methods. Then, because there is duplicate value(last element) the object array is decremented by 1.

Case 4: sample output deleting “Tris”



User chooses Player:

Player class

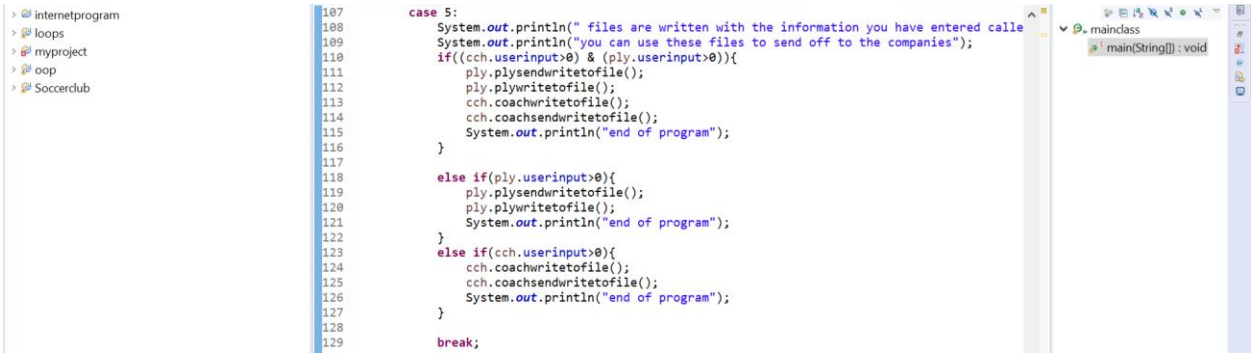
Deleting Player mimics Coach but with the extra attributes.

Case 4: sample output deleting “Cameron”

```
please enter in where you want to go 1. adding people 2. editing 3. printing out 4. deleting people 5. write to file 6.read from file
4
you will be deleting elements in the database
who do you want to edit, a coach or player
player
The name of player 1    Cameron
the jerseynumber for player 1   is 12
the jersey size of player 1     s
The jacket size of player 1     m
 the shortsize of player1       m
The name of player 2    Barron
the jerseynumber for player 2   is 10
the jersey size of player 2     s
The jacket size of player 2     s
 the shortsize of player2       s
which person do you want to delete
Cameron
name found
deleting has finished
 if you want to repeat 'yes'
```

Case 5: Performing write to file

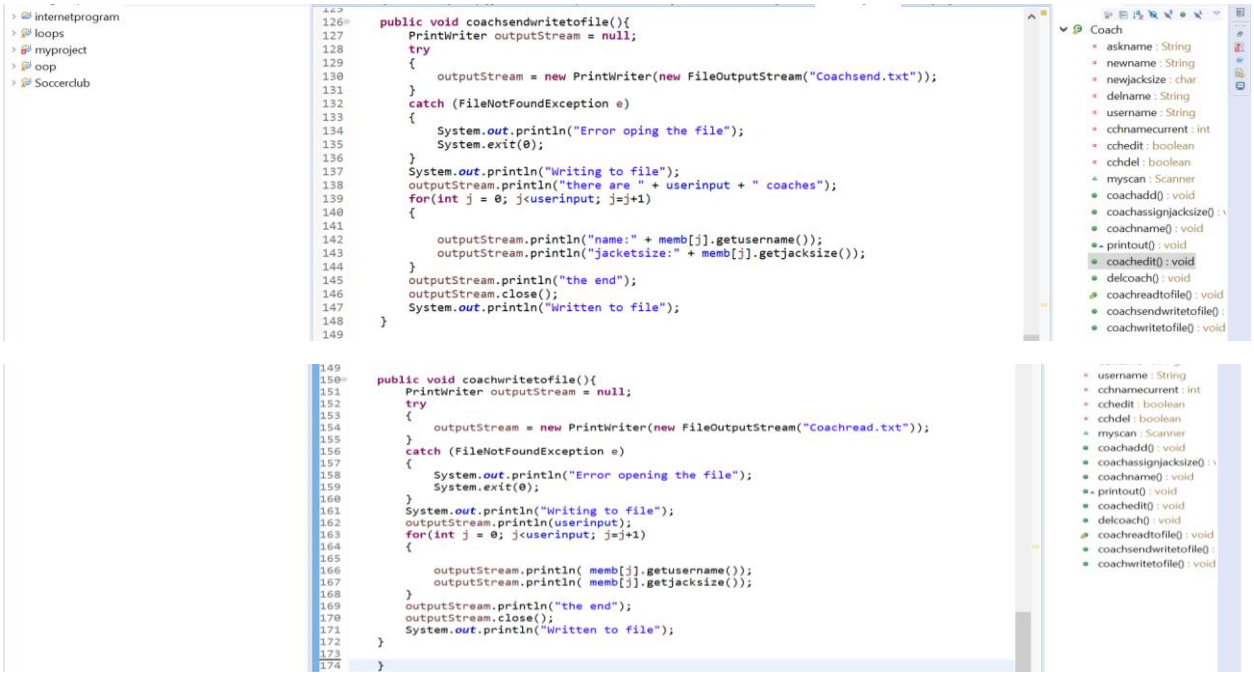
Mainclass:



The if statements evaluate only when userinput is greater than zero for either player or coach or both. This ensures that there are not any exceptions caught when running the program and is beneficial since the program will know if coaches or players are entered. Also, here there are two files created each for player and coach. One file is for sending off to companies, while the other is for reading back into the program. This was done so that a more formal presentation can be given on the file sent off to companies and the other can be generic and used for reading purposes.

If cch.userinput>0

Coach class



The try and catch method is used so that the program will catch the error and an error message is printed out. But, if the file opens successfully then all the attributes of the Player and Coach objects are written to file.

Sample files with coaches

Coachsend - Notepad

File Edit Format View Help

```
there are 2 coaches
name:Carl
jacketsize:s
name:Tris
jacketsize:l
the end
```

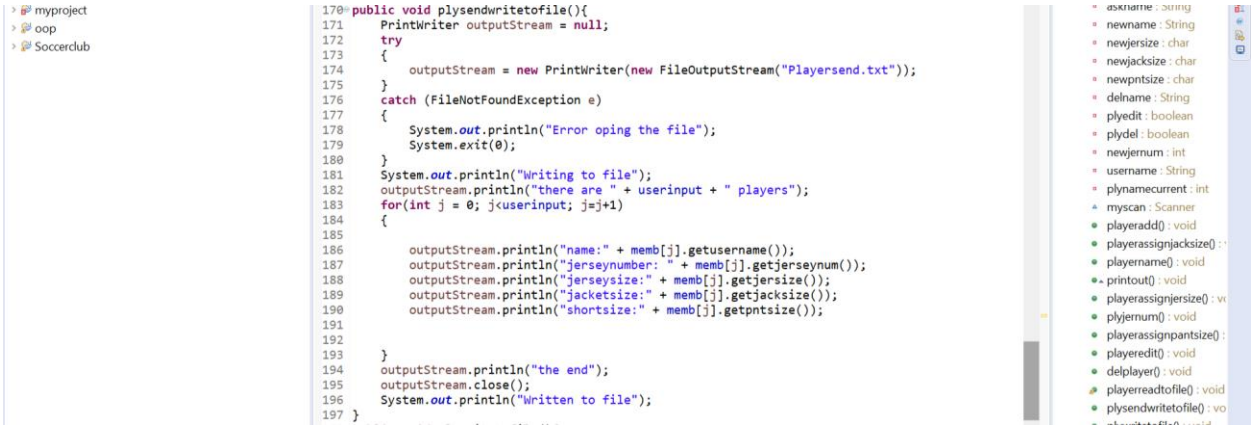
Coachread - Notepad

File Edit Format View Help

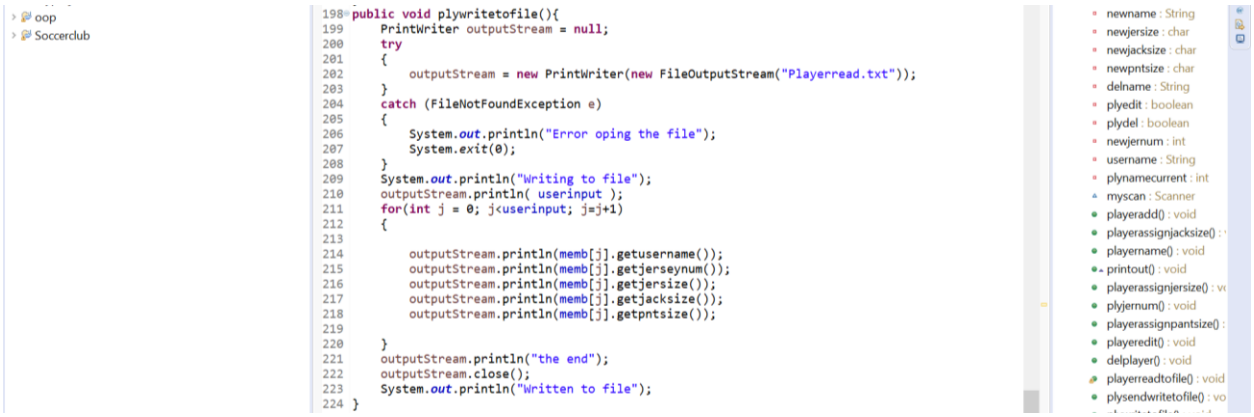
```
2
Carl
s
Tris
l
the end
```

If ply.userinput>0

Player class

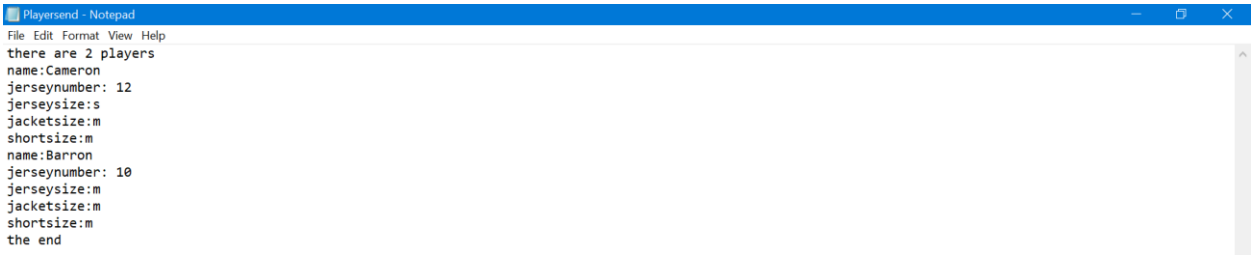


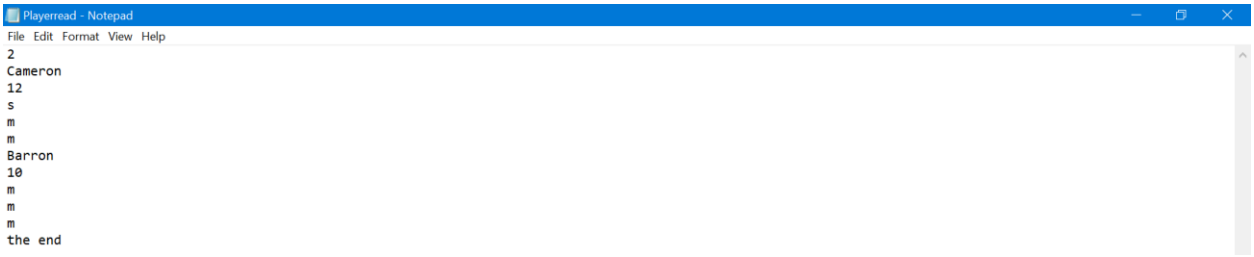
```
public void plysendwritetofile(){
    PrintWriter outputStream = null;
    try
    {
        outputStream = new PrintWriter(new FileOutputStream("Playersend.txt"));
    }
    catch (FileNotFoundException e)
    {
        System.out.println("Error oping the file");
        System.exit(0);
    }
    System.out.println("Writing to file");
    outputStream.println("there are " + userinput + " players");
    for(int j = 0; j<userinput; j=j+1)
    {

        outputStream.println("name:" + memb[j].getusername());
        outputStream.println("jerseynumber: " + memb[j].getjerseynum());
        outputStream.println("jerseysize:" + memb[j].getjersize());
        outputStream.println("jacketsize:" + memb[j].getjacksize());
        outputStream.println("shortsize:" + memb[j].getpntsize());


    }
    outputStream.println("the end");
    outputStream.close();
    System.out.println("Written to file");
}
```



```
public void plywritetofile(){
    PrintWriter outputStream = null;
    try
    {
        outputStream = new PrintWriter(new FileOutputStream("Playerread.txt"));
    }
    catch (FileNotFoundException e)
    {
        System.out.println("Error oping the file");
        System.exit(0);
    }
    System.out.println("Writing to file");
    outputStream.println( userinput );
    for(int j = 0; j<userinput; j=j+1)
    {

        outputStream.println(memb[j].getusername());
        outputStream.println(memb[j].getjerseynum());
        outputStream.println(memb[j].getjersize());
        outputStream.println(memb[j].getjacksize());
        outputStream.println(memb[j].getpntsize());

    }
    outputStream.println("the end");
    outputStream.close();
    System.out.println("Written to file");
}
```

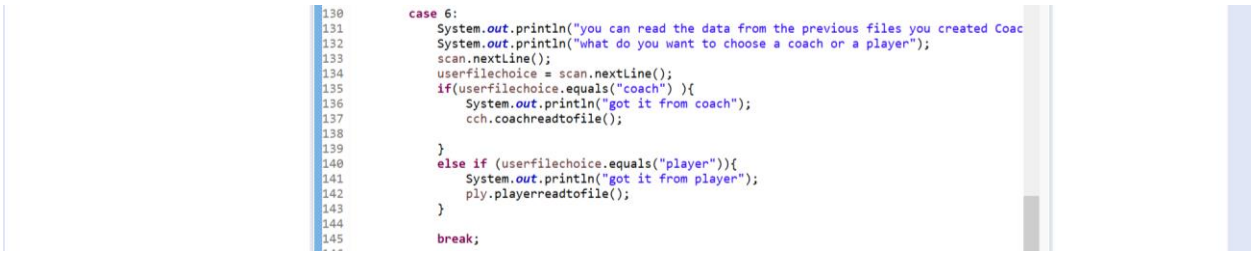
Sample files with players





Case 6: Performing read from file

Mainclass:

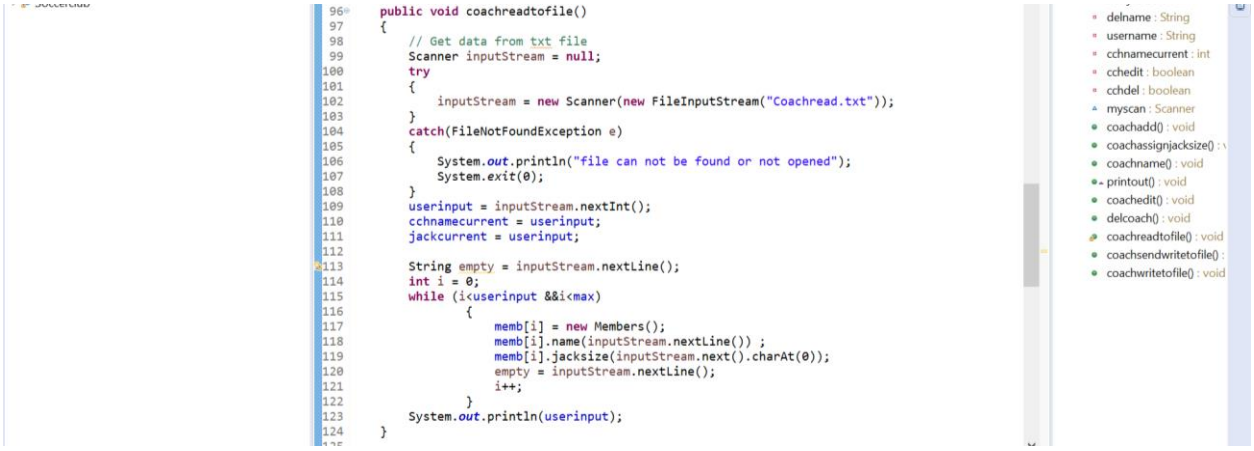


```
130         case 6:
131             System.out.println("you can read the data from the previous files you created Coac
132             System.out.println("what do you want to choose a coach or a player");
133             scan.nextLine();
134             userfilechoice = scan.nextLine();
135             if(userfilechoice.equals("coach") ){
136                 System.out.println("got it from coach");
137                 cch.coachreadtofile();
138 
139             }
140             else if (userfilechoice.equals("player")){
141                 System.out.println("got it from player");
142                 ply.playerreadtofile();
143             }
144 
145             break;
```

The read file evaluates based on whether the user wants to read from the coach file or the player file. Thus, there is importance of creating separate read files for both Coach and Player. Read file is beneficial because the user can return back to a previously written file and perform all the case statements (add, delete etc.) after reading from file.

User chooses coach:

Coach class:



```
public void coachreadtofile()
{
    // Get data from txt file
    Scanner inputStream = null;
    try
    {
        inputStream = new Scanner(new FileInputStream("Coachread.txt"));
    }
    catch(FileNotFoundException e)
    {
        System.out.println("file can not be found or not opened");
        System.exit(0);
    }
    userinput = inputStream.nextInt();
    cchnamecurrent = userinput;
    jackcurrent = userinput;

    String empty = inputStream.nextLine();
    int i = 0;
    while (i<userinput &&i<max)
            {
                memb[i] = new Members();
                memb[i].name(inputStream.nextLine()) ;
                memb[i].jacksize(inputStream.next().charAt(0));
                empty = inputStream.nextLine();
                i++;
            }
    System.out.println(userinput);
}
```

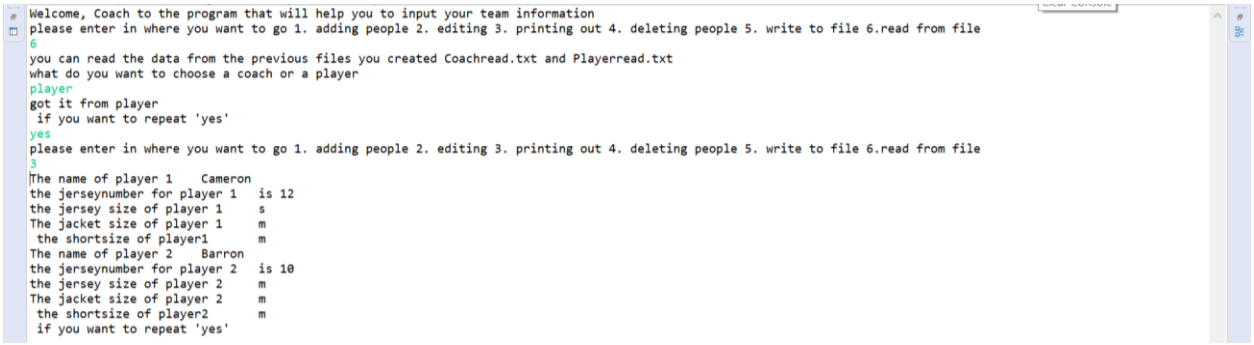
```
mainclass [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Mar 18, 2018, 6:29:17 PM)
Welcome, Coach to the program that will help you to input your team information
please enter in where you want to go 1. adding people 2. editing 3. printing out 4. deleting people 5. write to file 6.read from file
6
you can read the data from the previous files you created Coachread.txt and Playerread.txt
what do you want to choose a coach or a player
coach
got it from coach
2
 if you want to repeat 'yes'
yes
please enter in where you want to go 1. adding people 2. editing 3. printing out 4. deleting people 5. write to file 6.read from file
3
The name of Coach 1     Carl
The jacket size of Coach 1      s
The name of Coach 2     Tris
The jacket size of Coach 2      l
 if you want to repeat 'yes'
```

The example shows how after reading the file the program executes the printout(Case 3). The other case statements can also be executed based on the user's desires. Then, the new elements can be written back to file(Case 5) which rewrites all the files.

User chooses player:

Player class:

```
}
public void playerreadtofile()
{
    // Get data from txt file
    Scanner inputStream = null;
    try
    {
        inputStream = new Scanner(new FileInputStream("Playerread.txt"));
    }
    catch(FileNotFoundException e)
    {
        System.out.println("file can not be found or not opened");
        System.exit(0);
    }
    userinput = inputStream.nextInt();
    plynamecurrent = userinput;
    jackcurrent = userinput;
    jercurrent = userinput;
    pantcurrent = userinput;
    jerseynumcurrent = userinput;
    String empty = inputStream.nextLine();
    int i = 0;
    while (i<userinput &&i<max)
            {
                memb[i] = new Members();
                memb[i].name(inputStream.nextLine()) ;
                memb[i].jerseynumber(inputStream.nextInt());
                empty = inputStream.nextLine();
                memb[i].jersize(inputStream.next().charAt(0));
                memb[i].jacksize(inputStream.next().charAt(0));
                memb[i].pantsize(inputStream.next().charAt(0));
                empty = inputStream.nextLine();
                i++;
            }
}
```



Word Count: 903