

Criterion C: Development

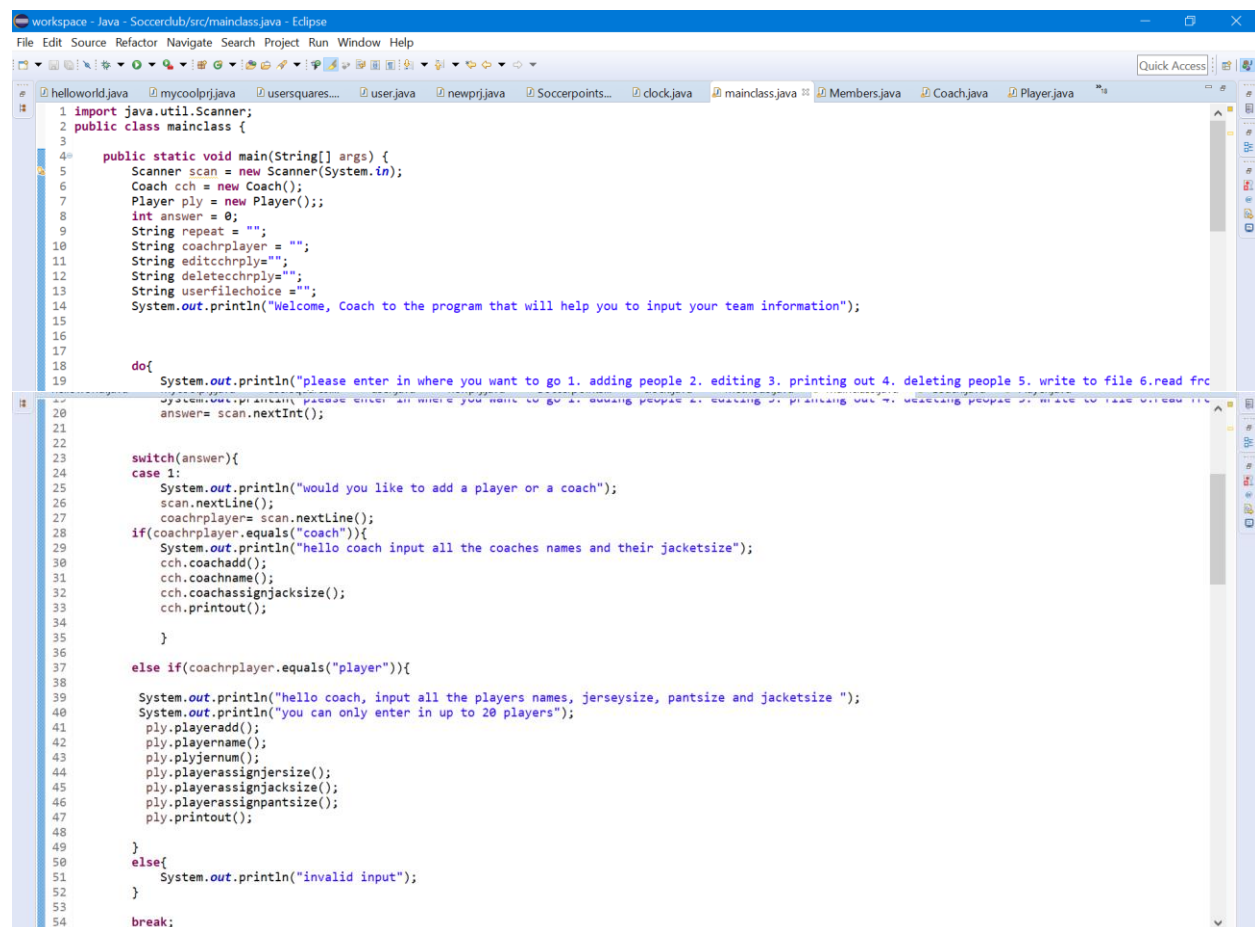
The Mainclass has six case statements performing functions of (add, edit, printout, delete, writetofile and read from file.

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 from file");
20         answer= scan.nextInt();
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Case 1: Performing the add function

Mainclass:



```
1 import java.util.Scanner;
2 public class mainclass {
3
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6         Coach cch = new Coach();
7         Player ply = new Player();
8         int answer = 0;
9         String repeat = "";
10        String coachrply = "";
11        String editcchply = "";
12        String deletecchply = "";
13        String userfilechoice = "";
14        System.out.println("Welcome, Coach to the program that will help you to input your team information");
15
16
17
18
19        do{
20            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 from file");
21            answer= scan.nextInt();
22
23
24            switch(answer){
25                case 1:
26                    System.out.println("would you like to add a player or a coach");
27                    scan.nextLine();
28                    coachrply= scan.nextLine();
29                    if(coachrply.equals("coach")){
30                        System.out.println("hello coach input all the coaches names and their jacketsizes");
31                        cch.coachadd();
32                        cch.coachname();
33                        cch.coachassignjacksizes();
34                        cch.printout();
35                    }
36                    else if(coachrply.equals("player")){
37
38
39                        System.out.println("hello coach, input all the players names, jerseysize, pantsize and jacketsize ");
40                        System.out.println("you can only enter in up to 20 players");
41                        ply.playeradd();
42                        ply.playername();
43                        ply.playerjersize();
44                        ply.playerassignjersize();
45                        ply.playerassignjacksizes();
46                        ply.playerassignpantsizes();
47                        ply.printout();
48                    }
49                }
50                else{
51                    System.out.println("invalid input");
52                }
53            }
54            break;
```

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 newjacksz;
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 coachassignjacksz(){
24        super.assignjacksz();
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].getjacksz());
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 coachassignjacksz() 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:

```
> internetprogram
> loops
> myproject
> oop
> Soccerclub

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 jerseyum;
14    private char jacksz;
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 jerseyumcurrent;
23    private String askname;
24    private String newname;
25    private String delname;
26    private char newjersize;
27    private char newjacksz;
28    Scanner myscan = new Scanner(System.in);
```

```
Members
+ max : int
+ memb : Members[]
+ i : int
+ username : String
+ jersize : char
+ jerseyum : int
+ jacksz : char
+ pntsize : char
+ namecurrent : int
+ jercurrent : int
+ jackcurrent : int
+ userinput : int
+ numof : int
+ pantcurrent : int
+ jerseyumcurrent : int
+ askname : String
+ newname : String
+ delname : String
```

```

33= public void addmember(){
34     System.out.println("how many people would you like to add");
35     numof = myscan.nextInt();
36     myscan.nextLine();
37     if ((numof + userInput) < memb.length){
38         System.out.println(numof + userInput);
39         for(int i = 0 ; i<numof; i++){
40             {
41                 memb[userinput] = new Members();
42                 userinput = userInput +1;
43             }
44         }
45     }
46     else
47     {
48         System.out.println(numof+" new members will exceed the number of legit members");
49     }
50 }
51= public void assignname(){
52     for(i =0; i<numof ; i++){
53         System.out.println("enter in the name of person " + (namecurrent+1));
54         username = myscan.nextLine();
55         memb[namecurrent].name(username);
56         namecurrent = namecurrent+1;
57     }
58 }
59 }
60 }
61= public void assignjersize(){
62     for(i =0; i<numof; i++){
63         System.out.println("type 's' for small, 'm' for medium and 'l' for large");
64         System.out.println("enter the jersize for " + memb[jcurrent].getusername());
65         jersize = myscan.next().charAt(0);
66         memb[jcurrent].jersize(jersize);
67         jcurrent = jcurrent+1;
68 }

```

Case1: Sample output with two coaches

```

20 mainclass [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Mar 18, 2018, 10:55:29 AM)
21 Welcome, Coach to the program that will help you to input your team information
22 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
23 1
24 would you like to add a player or a coach
25 coach
26 hello coach input all the coaches names and their jacketsize
27 how many people would you like to add
28 2
29 2
30 enter in the name of coach 1
31 Tom
32 enter in the name of coach 2
33 Tris
34 type 's' for small, 'm' for medium and 'l' for large
35 enter in the jacket size for Tom
36 m
37 type 's' for small, 'm' for medium and 'l' for large
38 enter in the jacket size for Tris
39 1
40 The name of Coach 1 Tom
41 The jacket size of Coach 1 m
42 The name of Coach 2 Tris
43 The jacket size of Coach 2 1
44 if you want to repeat 'yes'
45

```

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

Player class:

```

15 private int newjernum;
16 private String username;
17 private int plynamecurrent;
18 Scanner myscan = new Scanner(System.in);
19
20= public void playeradd(){
21     super.addmember();
22 }
23
24 }
25= public void playerassignjackszie(){
26     super.assignjackszie();
27 }
28= public void playername()
29 {
30     for(int i =0; i<numof ; i++){
31         System.out.println("enter in the name of player " + (plynamecurrent+1));
32         username = myscan.nextLine();
33         memb[plynamecurrent].name(username);
34         plynamecurrent = plynamecurrent+1;
35     }
36 }
37= public void printout()
38 {
39     for(int i = 0; i<userinput; i++){
40         System.out.println("The name of player " + (i+1) + '\t' + memb[i].getusername());
41         System.out.println("the jersey number for player " + (i+1) + '\t' + "is " + memb[i].getjersynum() );
42         System.out.println("the jersey size of player " + (i+1) + '\t' + memb[i].getjersize());
43         System.out.println("The jacket size of player " + (i+1) + '\t' + memb[i].getjackszie());
44         System.out.println(" the shorts size of player " + (i+1) + '\t' + memb[i].getpntszie());
45     }
46 }
47 }
48 }
49= public void playerassignjersize()
50 {
51     super.assignjersize();
52 }
53 }
54= public void plyjernum()
55 {
56     super.jersynummemb();
57 }
58= public void playerassignpantzsize()
59 {
60     super.assignpantzsize();
61 }
62 }

```

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 jersey num or jersey size 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 jersey number for " + memb[jerseynumcurrent].getusername());
76         jersey num = myscan.nextInt();
77         memb[jerseynumcurrent].jersey number(jersey num);
78         jersey numcurrent= jersey numcurrent+1;
79     }
80 }
81= public void assignjackszize(){
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         jackszize = myscan.next().charAt(0);
86         memb[jackcurrent].jackszize(jackszize);
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;
97     }
98 }
```

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, jersey size, pants size and jackets size
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 jersey number for Cameron
12
enter the jersey number 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 jersey number 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 jersey number 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'
```

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:

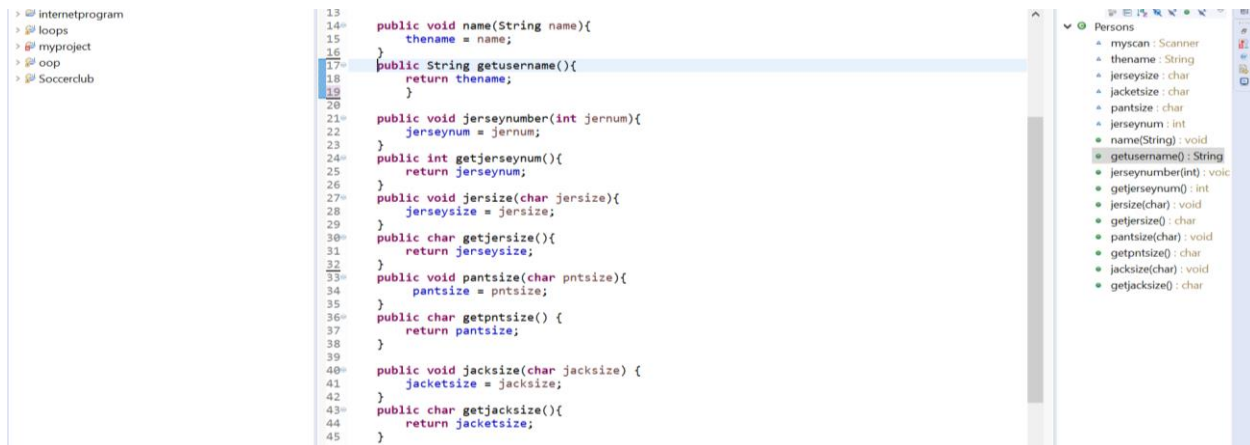
Coach class

```
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 jacksize to");
55                 newjacksize = myscan.next().charAt(0);
56                 memb[i].jacksize(newjacksize);
57                 cchedit = true;
58             }
59         }
60     }
61     if(cchedit){
62         System.out.println("editing has finished");
63     }
64     else{
65         System.out.println("invalid input");
66     }
67 }
```

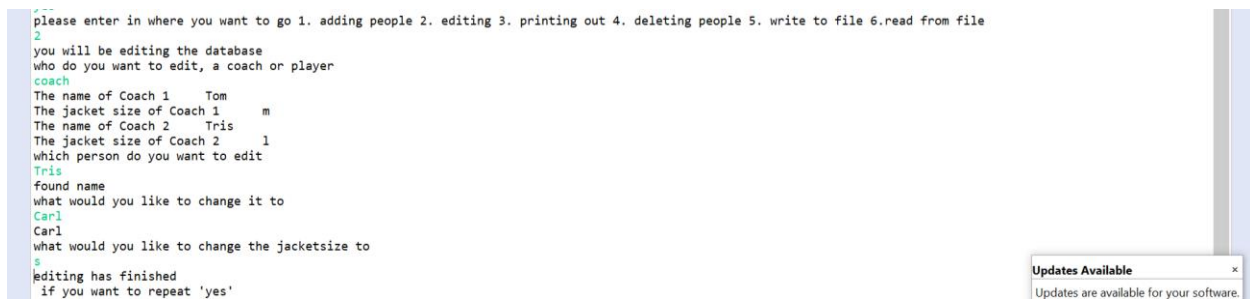
Updates Available
Updates are available for your software.
Click to review and install updates.

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

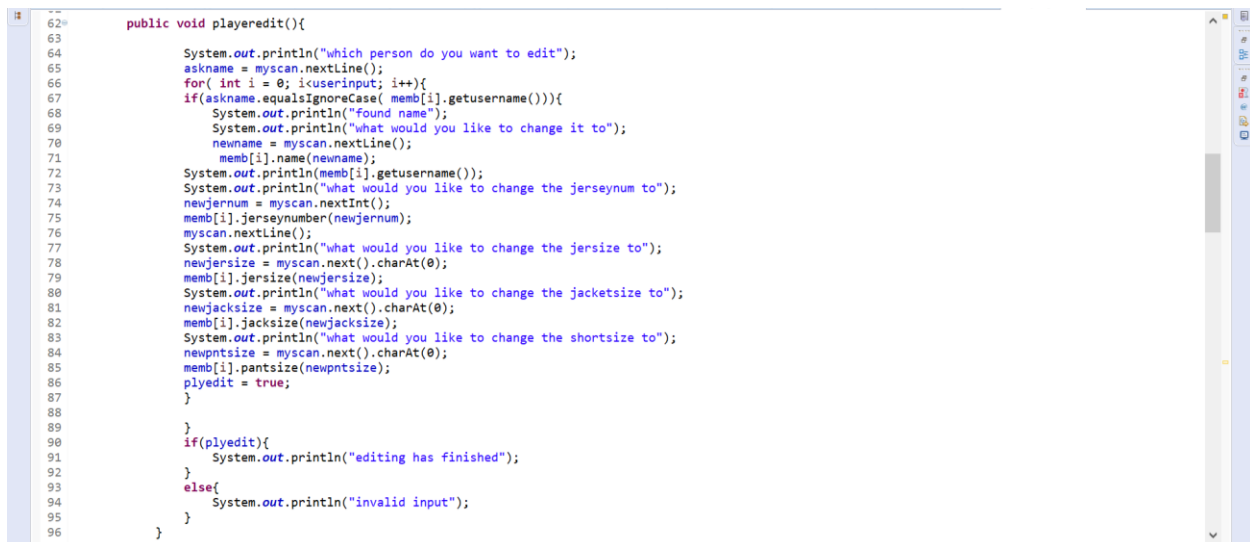


Case 2: Sample output with coach edit



User chooses player

Player Class



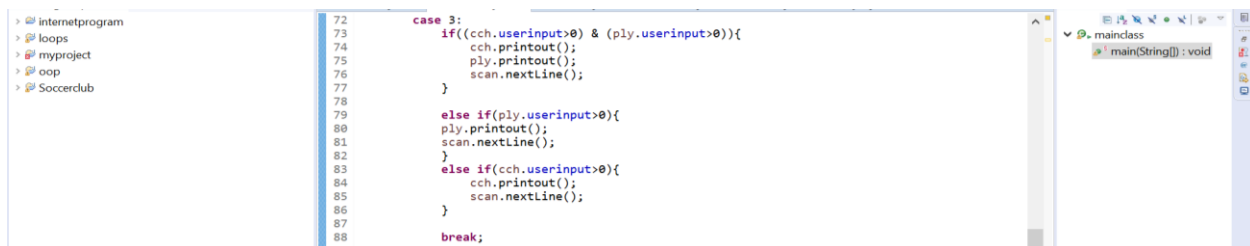
`Playeredit()` functions in the same way as `coachedit()` but, the difference being the editing of the extra attributes of the Player object. The Boolean `plyedit` functions the same as in Coach class.

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 jersey number 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 jersey number 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
what would you like to change the jersey num 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



```
72 case 3:
73     if((cch.userinput>0) & (ply.userinput>0)){
74         cch.printout();
75         ply.printout();
76         scan.nextLine();
77     }
78
79     else if(ply.userinput>0){
80         ply.printout();
81         scan.nextLine();
82     }
83
84     else if(cch.userinput>0){
85         cch.printout();
86         scan.nextLine();
87     }
88
89     break;
```

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

```
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].getjackszsize());
40     }
41 }
42
```

Player class

```
38 public void printout()
39 {
40     for(int i = 0; i<userinput; i++){
41         System.out.println("The name of player " + (i+1) + '\t' + memb[i].getusername());
42         System.out.println("the jersey number for player " + (i+1) + '\t' + "is " + memb[i].getjerseynum() );
43         System.out.println("the jersey size of player " + (i+1) + '\t' + memb[i].getjersize());
44         System.out.println("The jacket size of player " + (i+1) + '\t' + memb[i].getjackszsize());
45         System.out.println(" the shortsize of player" + (i+1) + '\t' + memb[i].getpntsizsize());
46     }
47 }
48
49
```

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

Case 4: Performing Delete

Mainclass

```
> internetprogram
> loops
> myproject
> oop
> Soccerclub

90 case 4:
91 System.out.println("you will be deleting elements in the database");
92 System.out.println("who do you want to edit, a coach or player");
93 scan.nextLine();
94 deletecchrply = scan.nextLine();
95 if(deletecchrply.equals("coach") & (cch.userinput>0)){
96     cch.printout();
97     cch.delcoach();
98 }
99 else if(deletecchrply.equals("player") & (ply.userinput>0)){
100     ply.printout();
101     ply.delplayer();
102 }
103 else{
104     System.out.println("you either do not have a coach or player entered, please a
105 }
106 break;
```

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

```
69 public void delcoach(){
70     String prevname;
71     char prevjacksize;
72     System.out.println("which person do you want to delete");
73     delname= myscan.nextLine();
74     for( int i = 0; i<userinput; i++){
75         if(delname.equalsIgnoreCase(memb[i].getusername())){
76             System.out.println("name found");
77             prevname = memb[userinput-1].getusername();
78             memb[i].name(prevname);
79             userinput = userinput-1;
80             prevjacksize = memb[jackcurrent-1].getjacksize();
81             memb[i].jacksize(prevjacksize);
82             jackcurrent= jackcurrent-1;
83             cchnamecurrent = cchnamecurrent-1;
84             cchdel = true;
85         }
86     }
87 }
88 if(cchdel){
89     System.out.println("deleting has finished");
90 }
91 else{
92     System.out.println("invalid input");
93 }
94 }
95 }
```

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”

```
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
coach
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
which person do you want to delete
Tris
name found
deleting has finished
if you want to repeat 'yes'
```

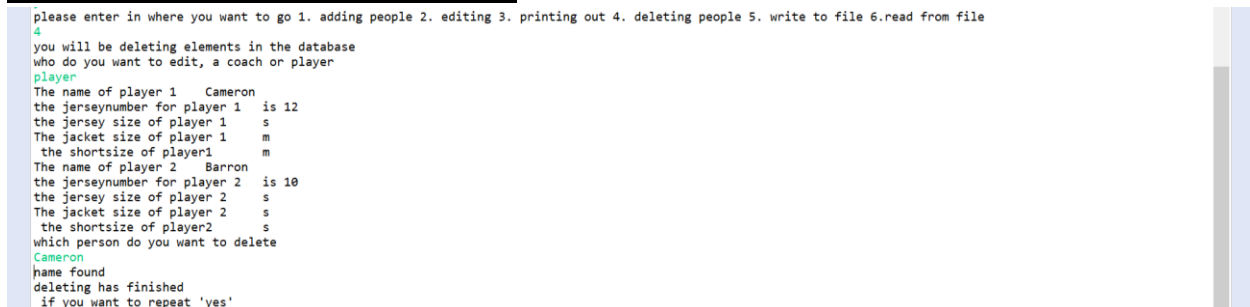

User chooses Player:

Player class



Deleting Player mimics Coach but with the extra attributes.

Case 4: sample output deleting "Cameron"



Case 5: Performing write to file

Mainclass:

```
107 case 5:
108     System.out.println(" files are written with the information you have entered call");
109     System.out.println("you can use these files to send off to the companies");
110     if((cch.userinput>0) & (ply.userinput>0)){
111         ply.plysendwritetofile();
112         ply.plywritetofile();
113         cch.coachwritetofile();
114         cch.coachsendwritetofile();
115         System.out.println("end of program");
116     }
117
118     else if(ply.userinput>0){
119         ply.plysendwritetofile();
120         ply.plywritetofile();
121         System.out.println("end of program");
122     }
123     else if(cch.userinput>0){
124         cch.coachwritetofile();
125         cch.coachsendwritetofile();
126         System.out.println("end of program");
127     }
128
129     break;
```

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

```
127 public void coachsendwritetofile(){
128     PrintWriter outputStream = null;
129     try
130     {
131         outputStream = new PrintWriter(new FileOutputStream("Coachsend.txt"));
132     }
133     catch (FileNotFoundException e)
134     {
135         System.out.println("Error oping the file");
136         System.exit(0);
137     }
138     System.out.println("Writing to file");
139     outputStream.println("there are " + userinput + " coaches");
140     for(int j = 0; j<userinput; j=j+1)
141     {
142         outputStream.println("name: " + memb[j].getusername());
143         outputStream.println("jacketsize: " + memb[j].getjacketsize());
144     }
145     outputStream.println("the end");
146     outputStream.close();
147     System.out.println("Written to file");
148 }
149
150 public void coachwritetofile(){
151     PrintWriter outputStream = null;
152     try
153     {
154         outputStream = new PrintWriter(new FileOutputStream("Coachread.txt"));
155     }
156     catch (FileNotFoundException e)
157     {
158         System.out.println("Error opening the file");
159         System.exit(0);
160     }
161     System.out.println("Writing to file");
162     outputStream.println(userinput);
163     for(int j = 0; j<userinput; j=j+1)
164     {
165         outputStream.println( memb[j].getusername());
166         outputStream.println( memb[j].getjacketsize());
167     }
168     outputStream.println("the end");
169     outputStream.close();
170     System.out.println("Written to file");
171 }
172
173
174 }
```

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

```
> myproject
> oop
> Soccerdub
```

```
170 public void plysendwritetofile(){
171     PrintWriter outputStream = null;
172     try
173     {
174         outputStream = new PrintWriter(new FileOutputStream("Playersend.txt"));
175     }
176     catch (FileNotFoundException e)
177     {
178         System.out.println("Error oping the file");
179         System.exit(0);
180     }
181     System.out.println("Writing to file");
182     outputStream.println("there are " + userInput + " players");
183     for(int j = 0; j<userinput; j=j+1)
184     {
185
186         outputStream.println("name:" + memb[j].getusername());
187         outputStream.println("jerseynumber:" + memb[j].getjerseynum());
188         outputStream.println("jerseysize:" + memb[j].getjersize());
189         outputStream.println("jacketsize:" + memb[j].getjacksie());
190         outputStream.println("shortsize:" + memb[j].getpntsize());
191
192     }
193     outputStream.println("the end");
194     outputStream.close();
195     System.out.println("Written to file");
196 }
197 }
```

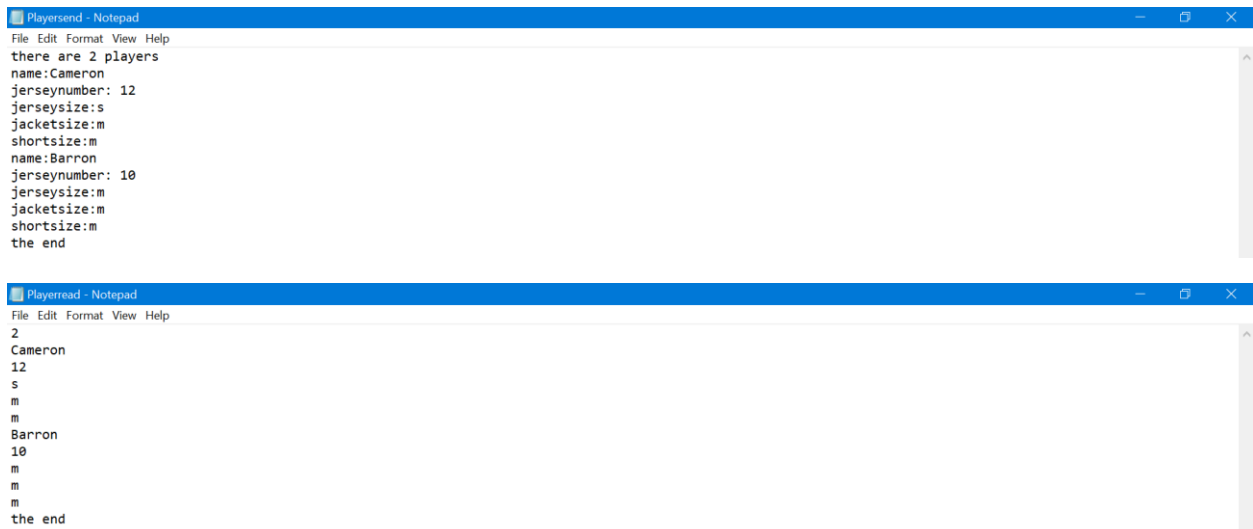
```
* exkname : String
* newname : String
* newjersize : char
* newjacksie : char
* newpntsize : char
* delname : String
* plyedit : boolean
* plydel : boolean
* newjernum : int
* username : String
* plynamecurrent : int
* myscan : Scanner
* playeradd() : void
* playerassignjacksie() : void
* playername() : void
* printout() : void
* playerassignjersize() : void
* plyjernum() : void
* playerassignpntsize() : void
* playeredit() : void
* delplayer() : void
* playerreadtofile() : void
* plysendwritetofile() : void
```

```
> oop
> Soccerdub
```

```
198 public void plywritetofile(){
199     PrintWriter outputStream = null;
200     try
201     {
202         outputStream = new PrintWriter(new FileOutputStream("Playerread.txt"));
203     }
204     catch (FileNotFoundException e)
205     {
206         System.out.println("Error oping the file");
207         System.exit(0);
208     }
209     System.out.println("Writing to file");
210     outputStream.println( userInput );
211     for(int j = 0; j<userinput; j=j+1)
212     {
213
214         outputStream.println(memb[j].getusername());
215         outputStream.println(memb[j].getjerseynum());
216         outputStream.println(memb[j].getjersize());
217         outputStream.println(memb[j].getjacksie());
218         outputStream.println(memb[j].getpntsize());
219
220     }
221     outputStream.println("the end");
222     outputStream.close();
223     System.out.println("Written to file");
224 }
```

```
* newname : String
* newjersize : char
* newjacksie : char
* newpntsize : char
* delname : String
* plyedit : boolean
* plydel : boolean
* newjernum : int
* username : String
* plynamecurrent : int
* myscan : Scanner
* playeradd() : void
* playerassignjacksie() : void
* playername() : void
* printout() : void
* playerassignjersize() : void
* plyjernum() : void
* playerassignpntsize() : void
* playeredit() : void
* delplayer() : void
* playerreadtofile() : void
* plysendwritetofile() : void
```

Sample files with players



The image shows two Notepad windows. The top window, titled 'Playersend - Notepad', contains the following text: 'there are 2 players', 'name:Cameron', 'jerseynumber: 12', 'jerseysize:s', 'jacketsize:m', 'shortsize:m', 'name:Barron', 'jerseynumber: 10', 'jerseysize:m', 'jacketsize:m', 'shortsize:m', and 'the end'. The bottom window, titled 'Playerread - Notepad', contains the following text: '2', 'Cameron', '12', 's', 'm', 'm', 'Barron', '10', 'm', 'm', 'm', and 'the end'.

```
Playersend - Notepad
File Edit Format View Help
there are 2 players
name:Cameron
jerseynumber: 12
jerseysize:s
jacketsize:m
shortsize:m
name:Barron
jerseynumber: 10
jerseysize:m
jacketsize:m
shortsize:m
the end

Playerread - Notepad
File Edit Format View Help
2
Cameron
12
s
m
m
Barron
10
m
m
m
the end
```

Case 6: Performing read from file

Mainclass:

```
130
131 case 6:
132     System.out.println("you can read the data from the previous files you created Coach
133     System.out.println("what do you want to choose a coach or a player");
134     scan.nextLine();
135     userfilechoice = scan.nextLine();
136     if(userfilechoice.equals("coach")){
137         System.out.println("got it from coach");
138         cch.coachreadtofile();
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:

The screenshot displays an IDE with two main components: a code editor on the left and a console window on the right.

Code Editor: The code is for a method named `coachreadtofile()`. It starts by initializing a `Scanner` for `Coachread.txt`. A `try-catch` block handles `FileNotFoundException`. If successful, it reads the file line by line, creating `Members` objects and storing them in an array. It also reads the jacket size for each coach. The code includes a `while` loop to process the input and a `System.out.println` statement to display the results.

Console Window: The output shows the program's execution. It starts with a welcome message and a menu. The user selects option 6 (read from file). The program then prompts the user to choose a coach or a player. The user enters 'coach'. The program then prompts the user to repeat the process. The user enters 'yes'. The program then displays the data for Coach 1 and Coach 2, including their names and jacket sizes.

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:

The screenshot displays an IDE with two main components: a code editor on the left and a console window on the right.

Code Editor: The code is for a method named `playerreadtofile()`. It starts by initializing a `Scanner` for `Playerread.txt`. A `try-catch` block handles `FileNotFoundException`. If successful, it reads the file line by line, creating `Members` objects and storing them in an array. It also reads the jersey number and jersey size for each player. The code includes a `while` loop to process the input and a `System.out.println` statement to display the results.

Console Window: The output shows the program's execution. It starts with a welcome message and a menu. The user selects option 6 (read from file). The program then prompts the user to choose a coach or a player. The user enters 'player'. The program then prompts the user to repeat the process. The user enters 'yes'. The program then displays the data for Player 1 and Player 2, including their names and jersey sizes.

```
# 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
player
# got it from player
# 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 player 1   Cameron
# the jersey number 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 jersey number 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'
```

Word Count: 903