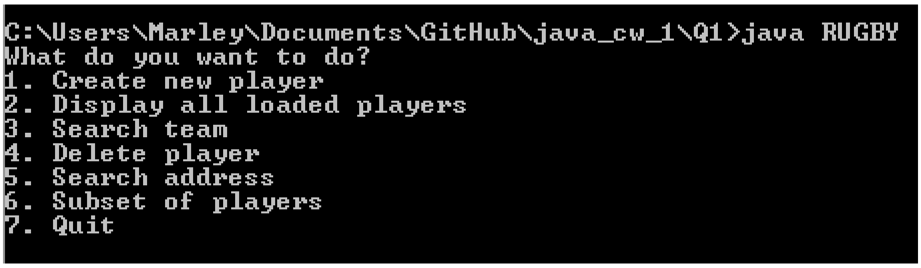
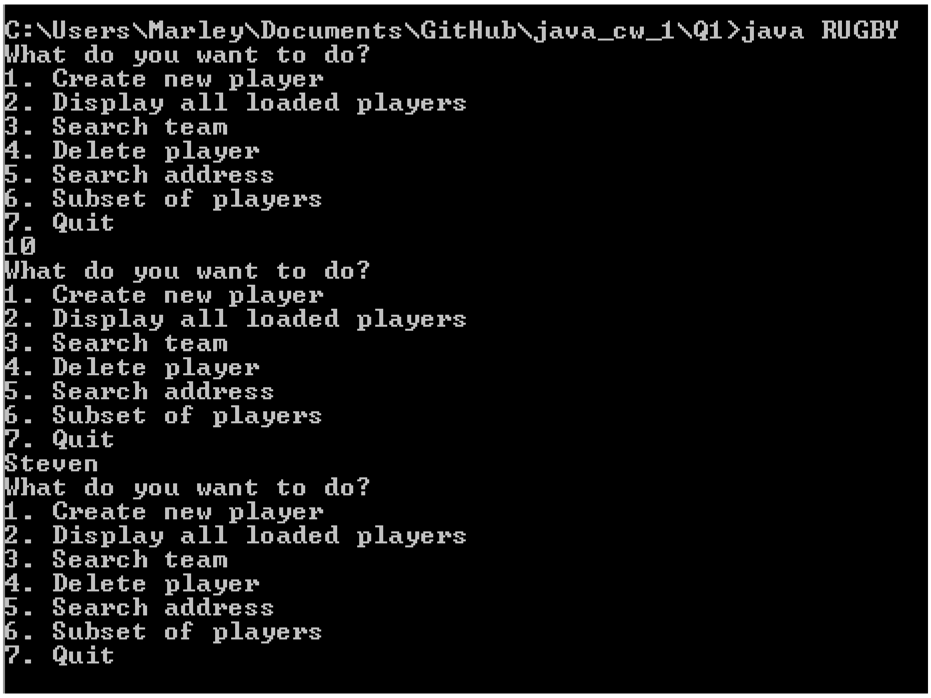


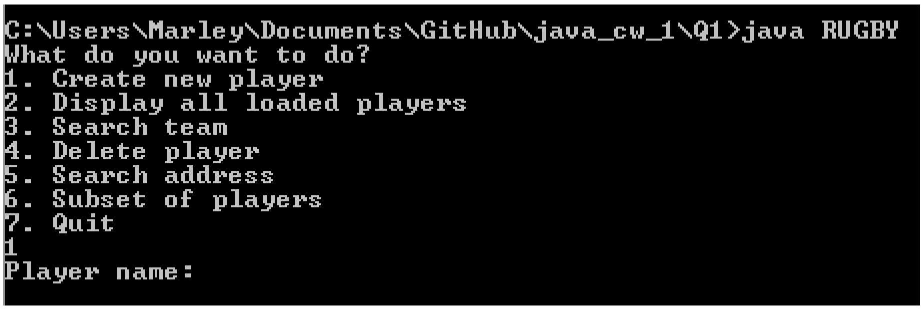
```
C:\Users\Marley\Documents\GitHub\java_cw_1\Q1>java RUGBY
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
```

This is the main menu of my program. By entering a number from 1-7 the user can select which item they want.



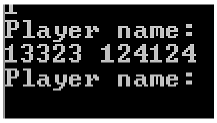
```
C:\Users\Marley\Documents\GitHub\java_cw_1\Q1>java RUGBY
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
10
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
Steven
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
```

Invalid answers are handled by displaying the menu again.



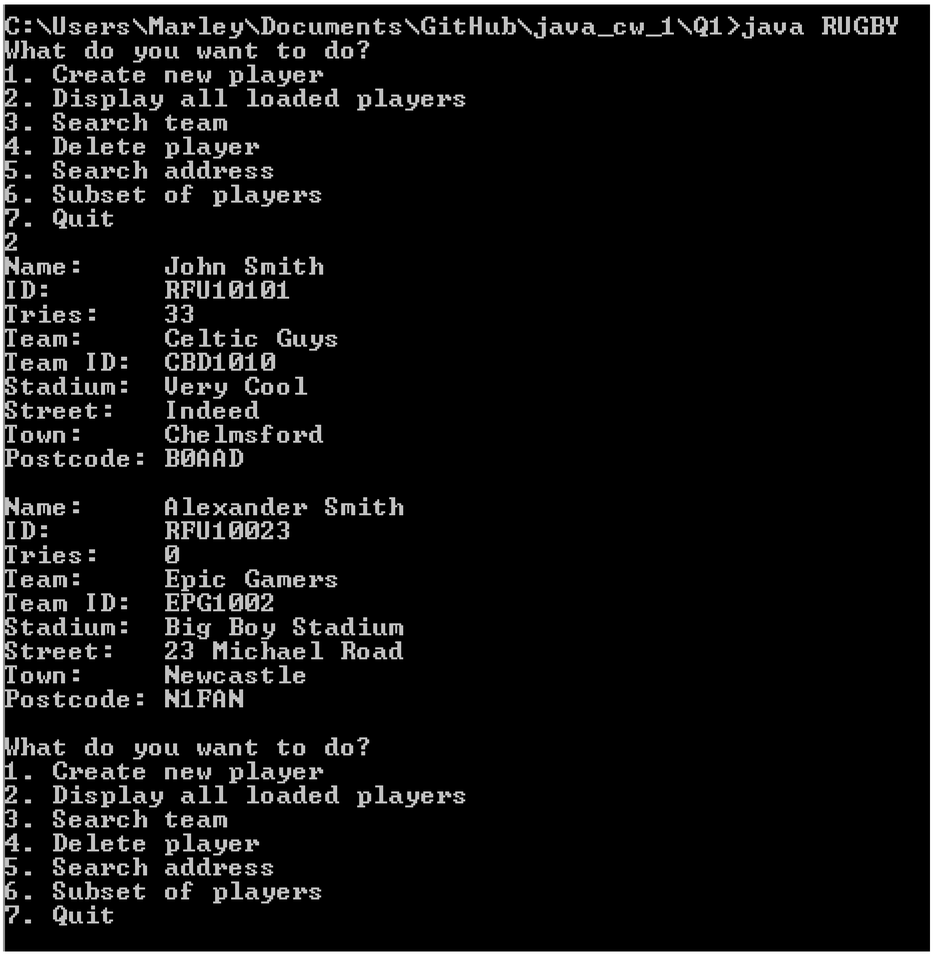
```
C:\Users\Marley\Documents\GitHub\java_cw_1\Q1>java RUGBY
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
1
Player name:
```

If the user selects to create a new player, they are individually prompted for each value.



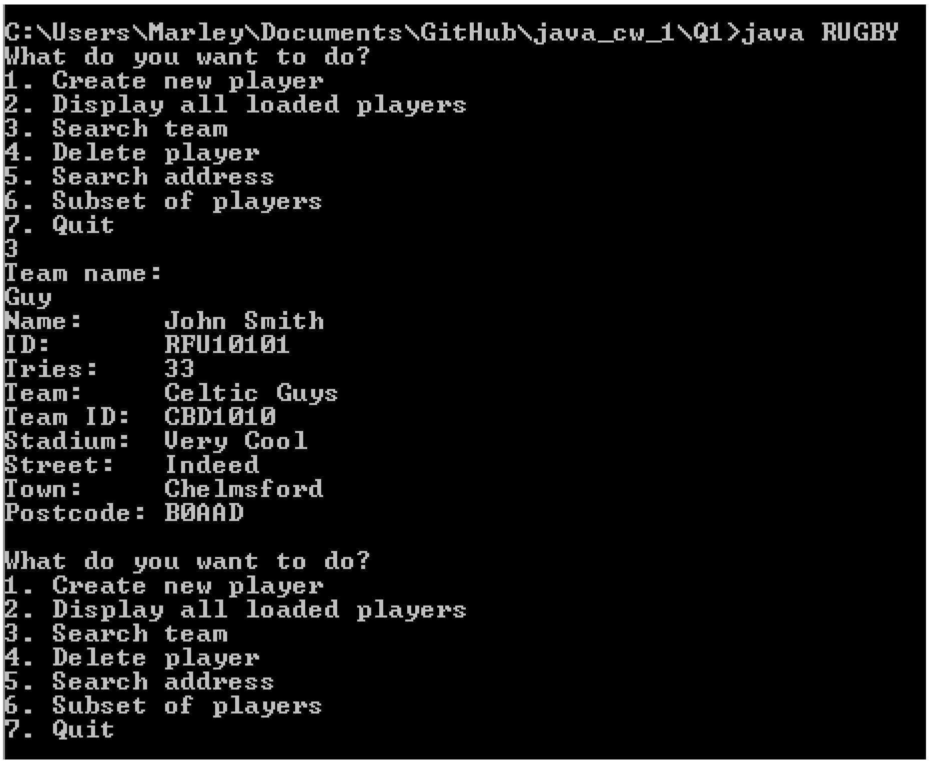
```
1
Player name:
13323 124124
Player name:
```

If the user enters invalid data then they are prompted again.



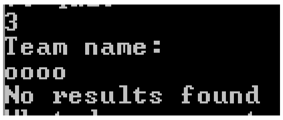
```
C:\Users\Marley\Documents\GitHub\java_cw_1\Q1>java RUGBY
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
2
Name:     John Smith
ID:       RFU10101
Tries:    33
Team:     Celtic Guys
Team ID:  CBD1010
Stadium:  Very Cool
Street:   Indeed
Town:     Chelmsford
Postcode: B0AAD

Name:     Alexander Smith
ID:       RFU10023
Tries:    0
Team:     Epic Gamers
Team ID:  EPG1002
Stadium:  Big Boy Stadium
Street:   23 Michael Road
Town:     Newcastle
Postcode: N1FAN

What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
```

If the user selects to display all loaded players, the full list of attributes for each player is displayed. It is the order of this list that should be considered when using the ‘Delete player’ command.



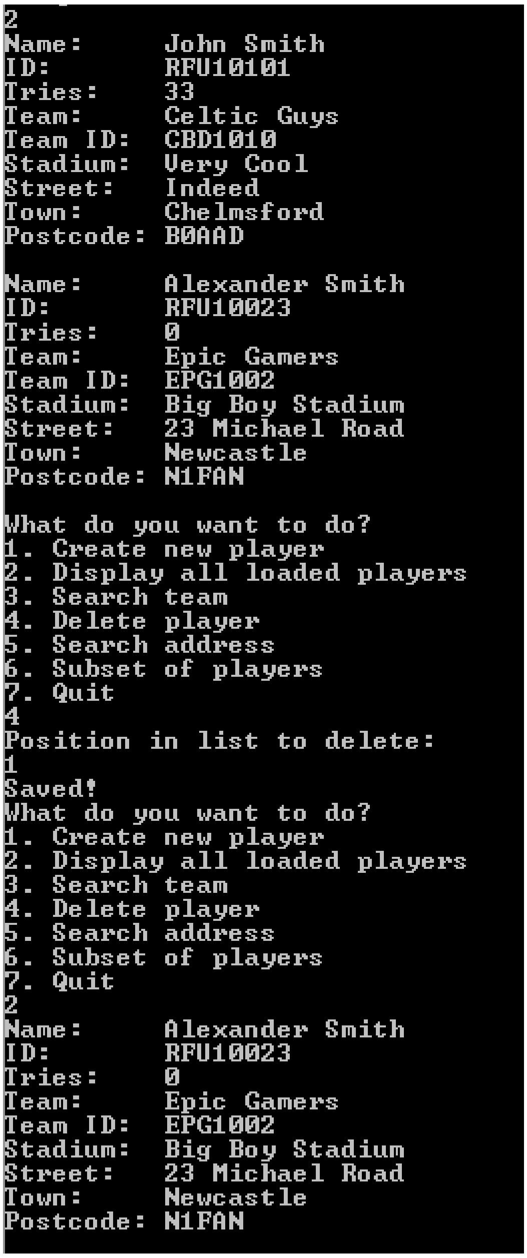
```
C:\Users\Marley\Documents\GitHub\java_cw_1\Q1>java RUGBY
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
3
Team name:
Guy
Name:     John Smith
ID:       RFU10101
Tries:    33
Team:     Celtic Guys
Team ID:  CBD1010
Stadium:  Very Cool
Street:   Indeed
Town:     Chelmsford
Postcode: B0AAD

What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
```

If the user chooses to search by team it displays all players whose team name contains the specified substring (case sensitive).



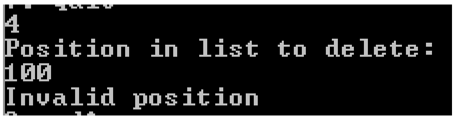
```
3
Team name:
oooo
No results found
```

The user is informed if no results are found.



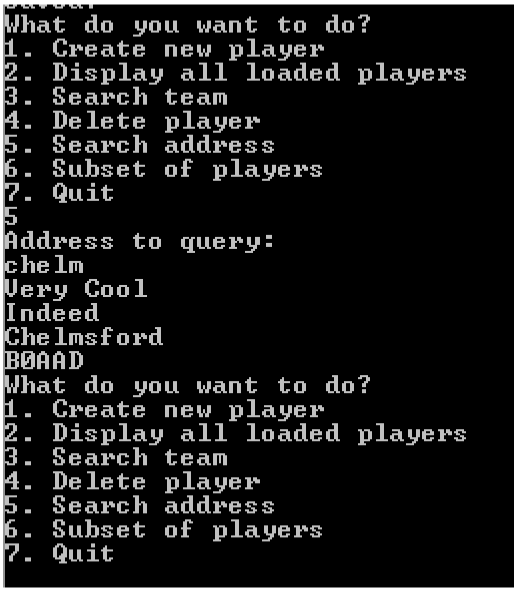
```
2
Name:     John Smith
ID:       RFU10101
Tries:    33
Team:     Celtic Guys
Team ID:  CBD1010
Stadium:  Very Cool
Street:   Indeed
Town:     Chelmsford
Postcode: B0AAD

Name:     Alexander Smith
ID:       RFU10023
Tries:    0
Team:     Epic Gamers
Team ID:  EPG1002
Stadium:  Big Boy Stadium
Street:   23 Michael Road
Town:     Newcastle
Postcode: N1FAN

What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
4
Position in list to delete:
1
Saved!
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
2
Name:     Alexander Smith
ID:       RFU10023
Tries:    0
Team:     Epic Gamers
Team ID:  EPG1002
Stadium:  Big Boy Stadium
Street:   23 Michael Road
Town:     Newcastle
Postcode: N1FAN
```

If the user chooses to delete a player they are then prompted for the position in list. Although the array that stores players is 0-indexed, the user is expected to use 1 for first, 2 for second etc.



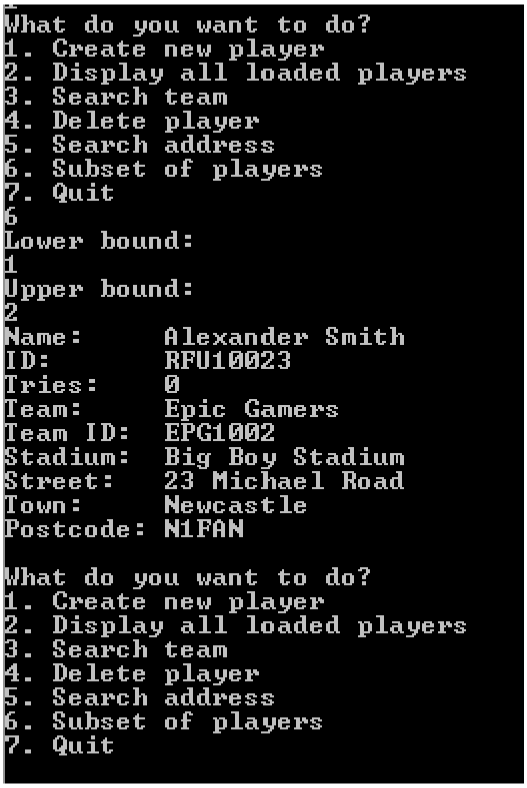
```
4
Position in list to delete:
100
Invalid position
```

If the user inputs an invalid position, they are notified.



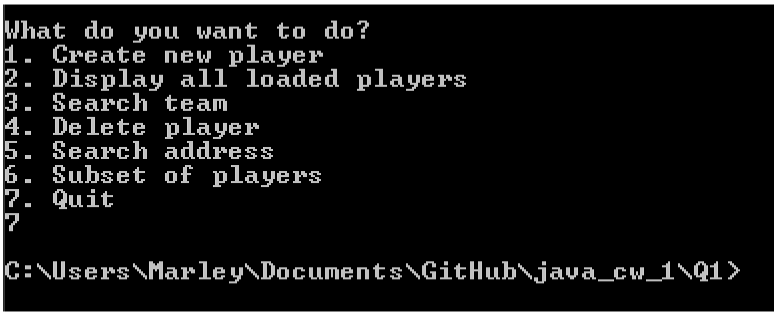
```
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
5
Address to query:
chelm
Very Cool
Indeed
Chelmsford
B0AAD
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
```

If the user chooses to search for an address they are prompted for a string to query. If the string matches any address element of any player, that address is then printed to the screen. The search is case-insensitive.



```
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
6
Lower bound:
1
Upper bound:
2
Name:     Alexander Smith
ID:       RFU10023
Tries:    0
Team:     Epic Gamers
Team ID:  EPG1002
Stadium:  Big Boy Stadium
Street:   23 Michael Road
Town:     Newcastle
Postcode: N1FAN

What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
```

If the user chooses to view a subset of players, the enter the lower bound (inclusive) and upper bound (exclusive) and then all players in that range are displayed. These bounds are from the perspective of the zero-indexed array.



```
What do you want to do?
1. Create new player
2. Display all loaded players
3. Search team
4. Delete player
5. Search address
6. Subset of players
7. Quit
7

C:\Users\Marley\Documents\GitHub\java_cw_1\Q1>
```

By entering 7 the user can exit the program.