```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;

public class GolfTournament {
      private int score_1;
      private int score_2;
      private String name;

      Scanner sc = new Scanner(System.in);
      public static void main(String[] args) {
           System.out.println("It's the Master's GolfTournament. Time
to track scores for the golfers.");
           System.out.println();

      System.out.println("_____________________________________________
_________________________________");
           GolfTournament object1 = new GolfTournament();
           System.out.println("Please enter the number of players: ");
           Scanner a1 = new Scanner(System.in);
           int userEntry = a1.nextInt();
           if(userEntry != 0) {
                object1.validate1(0, 40, userEntry);

                int[] firstScore = new int[userEntry];
                int[] secondScore = new int[userEntry];
                String[] playerId = new String[userEntry];
                int[] firstSecondSum = new int[userEntry];

                for(int i = 0; i < userEntry; i++) {
                      System.out.println("--------------------------
Player " + (i+1) +"-----------------------------");
                      playerId[i] = object1.playerNameMethod();
                      firstScore[i] = object1.playerFirstScore();
                      secondScore[i] = object1.playerSecondScore();



                }

                firstSecondSum = object1.combinedScore(firstScore,
secondScore);
                object1.disp(userEntry, playerId, firstScore,
secondScore, firstSecondSum);
                a1.close();
           }
           else if(userEntry == 0){
                System.out.println("Goodbye");
           }
```

```
    }
    //I preferred getters and setters instead of constructors inorder
to be more flexible
    public void setScore_1(int score_1) {
        this.score_1 = score_1;
    }

    public int getScore_1() {
        return score_1;
    }

    public void setScore_2(int score_2) {
        this.score_2 = score_2;
    }

    public int getScore_2() {
        return score_2;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void validate1(int a, int b, int userInput) {


        while((userInput < a) || (userInput > b)) {
            System.out.println("Please enter a valid number
between " + a + " and " + b + "\n");
            System.out.println("Please enter the number of
players: ");
            userInput = sc.nextInt();

            if((userInput > a) || (userInput < b)) {
                break;
            }
            else {
                continue;
            }
        }
    }

    public void validateResponse(int a, int b, int userInput, int
day) {

        while((userInput < a) || (userInput > b)) {
            System.out.println("Please enter a valid number
between " + a + " and " + b + "\n");
                  System.out.println("Please enter the day " + day + "
score:");
                  userInput = sc.nextInt();
            }
      }

      public int[] combinedScore(int[] arr1, int[] arr2) {
            int a = arr1.length;
            int[] combSum = new int[a];
            for(int i=0; i < a; i++) {
                  combSum[i] = arr1[i] + arr2[i];
            }

            return combSum;
      }

      public String playerNameMethod() {
            Scanner in = new Scanner(System.in);
            System.out.println("Please enter the player name: ");
            setName(in.nextLine());

            return this.getName();
      }

      public int playerFirstScore() {
            GolfTournament object2 = new GolfTournament();
            System.out.println("Please enter the day 1 score: ");
            int a = sc.nextInt();
            object2.validateResponse(0, 90, a, 1);
            setScore_1(a);

            return this.getScore_1();
      }

      public int playerSecondScore() {
            GolfTournament object3 = new GolfTournament();
            System.out.println("Please enter the day 2 score: ");
            int a = sc.nextInt();
            object3.validateResponse(0,  90, a, 2);
            setScore_2(a);

            return this.getScore_2();
      }

      public void disp(int num_rows, String[] arr1, int[] arr2, int[]
arr3, int[] arr4) {
            GolfTournament object5 = new GolfTournament();
            System.out.println("---------------------------------------
-----Tournament Statistics----------------------------------------");
            System.out.println("PlayerID"+"              " + "Day1
Score" +"        " + "Day2 Score" + "          " + "Combined Score" +
"\n");

           for(int i = 0; i < num_rows; i++) {
                System.out.println(arr1[i]+"                      " +
arr2[i] +"                        " + arr3[i] + "
" + arr4[i] );
           }
           System.out.println("\n-------------------------------------
------Leader--------------------------------------------------------
");
           System.out.println(object5.leader(arr1, arr4));

           object5.playersStroke(num_rows, 10, arr1, arr4);

           System.out.println("\n---------------------------------See
Ya' At the 19th Hole -------------------------------------------------
");
     }

     public void playersStroke(int rowNum, int strokeNum,  String[]
arr1, int[] arr2) {//This is where you left yesterday
           /*playersStroke Algorithm Author: Obed Nuertey
            * To get the list of players within 10 strokes:
            * 1. first use a for loop to sort arr2
            * store the least value to a variable; this is the
leader's score
            *
            * 2. use a for loop to loop through arr2 the number of
arr2 length times
            * use an if statement to verify if the numbers are within
a range of ten from the leader's score
            * this is if((arr2[i] >= a)||(arr2[i] <= a + strokeNum)
strokeNum happens to be 10
            * we store the index of the numbers obtained from the if
statement in a new array say st10
            *
            * 3. To get the names of those players within 10 strokes:
            * Of course we use another for loop
            * inside the for loop, we print the names of the players
within 10 strokes by; arr1[st10]
            * Done:
            *
            * Note: The above algorithm something is just my RoughWork
o. In the actual code I'll make some amendments
            * */
           GolfTournament object8 = new GolfTournament();
           //No.1 Solved
           String leader = object8.leader(arr1, arr2);  //Create a
string variable to store the leader's name
           int leaderIndex = Arrays.asList(arr1).indexOf(leader);
```

```
//Get the index of the leader
           int leaderScore = arr2[leaderIndex]; //Get the leader's
score and store it in a variable

           //No.2 Solved -1
           List<Integer> strokeIndex = new ArrayList<Integer>();

           for(int i = 0; i < rowNum; i++) {
                 //a = arr2[i];
                 if((arr2[i] > leaderScore)&&(arr2[i] < (leaderScore +
strokeNum))) {
                      strokeIndex.add(indexGetter(arr2,
arr2[i]));//problem 1
                 }
           }

           //No.3 Solved
           List<String> strokeNames = new ArrayList<String>();
           for(int i = 0; i < strokeIndex.size(); i++) {
                 strokeNames.add(arr1[strokeIndex.get(i)]);
           }

           //Print the name of those within the stroke
           System.out.println("------------------------------------
Players within " + strokeNum + " Strokes of leader--------------------
--------------\n");
           for(int i = 0; i<strokeNames.size(); i++) {

                 System.out.println(strokeNames.get(i) + "\n");
           }

     }
     // -1 change stokeIndex into ArrayList
     public String leader(String[] arr1, int[] arr2) {
           /*leader Algorithm Author: Obed Nuertey
            *To get the leader name;
            *Create a new integer array say newComb
            *Copy the content of arr2 into newComb
            *Sort the newComb array
            *Get the first element say 'a'. This is the maximum
element in the array
            *Now get the index of 'a' in arr2. This will be the index
linking the name of the leader.
            *Now get the name of the leader by getting the element of
the index from arr2.
            *Store the name in a string variable and return it.
            *Done.
            *
            *Note: The above algorithm something is just my RoughWork
o. In the actual code I'll make some amendments
            * */
```

```
}
            GolfTournament object10 = new GolfTournament();
            int temp, leastNumIndex;
            int leastNum;
            String leaderName;

            //Get the least score
            int least = arr2[arr2.length -1];

            for(int i = 0; i < arr2.length; i++) {
                  if(arr2[i] < least) {
                        least = arr2[i];
                  }
            }

            leastNum = least; //Store least number to variable leastNum

            leastNumIndex = object10.indexGetter(arr2, leastNum); //Get
the index of the maximum number from the original array //problem 2

            leaderName = arr1[leastNumIndex];

            return leaderName;
      }


      //I created this method to get indices because java lacks that
feature
      public int indexGetter(int[] arr1, int element) {
            //GolfTournament object10 = new GolfTournament();

            int[] indexArr1 = new int[arr1.length];

            for(int i = 0; i < arr1.length; i++) {
                  indexArr1[i] = i;
            }

            //indexArr1 gets elements from 0 to length of arr1

            int a = 0;

            for(int i = 0; i < arr1.length; i++) {
                  if(arr1[i] == element) {
                        a = a + indexArr1[i];
                  }
            }

            return a;
      }
}
```