Adam Cankaya
June 9, 2013
CS 161
Assignment #5

1. Understanding the problem

I need to write a program that can read in two text files containing strings of integers that can be used to simulate games of golf and print out a table of the results. The program needs to be able to read in the strings and convert them to integers properly, assign them to variables for the golf course itself and easy player, and then perform various calculations such as pars or overall lowest score. The program then needs to be able to nicely print out to the screen a table summarizing each player's score on each hole of the game along with some text stating whom is the winning player,

2. Devising a Plan/Design

I plan to have a simple main method which will call various other methods that when put together result in a game of golf being simulated. I will have a Course method to take the input file and generate a golf course with pars, a Player method to take the player score input in and assign it to an array of scores, a Scores method that will calculate various player totals and print out a nice table of the course results, and then a Winner method that will determine which player had the lowest score and print out some text stating as such.

Course()

```
{
        try
        {
                Scanner myFileReader = new Scanner(courseText);

                for( i < numHoles)
                        coursePars[i] = myFileReader.nextInt();
        }
        catch
        {
                System.out.println("There was an input error with the course pars file – try again.");
        }
}
```

Player()

```
{
        try
        {
                Scanner myFileReader = new Scanner(playerText);

                numPlayers = myFileReader.nextInt();

                for( i < numHoles)
                        scoresString = myFileReader.nextLine();
                                for(j < length. scoresString)
                                scoresStringArray[j] = scoresString.atChar[j]
                                playerScores[i+j] = Integer.parseInt(scoreString[j]);
        }
        catch
        {
                System.out.println(“There was an input error with the player pars file – try again.”);
        }
}
```

Scores()

```
        System.out.println("Hole # \t Par    Player 1 Score  Player 2 Score  Player 3 Score");

        for(i < numPlayers*numOfHoles; i = i + numPlayers)
                courseParTotal = courseParTotal + coursePars[holeNum-1];
                playerOneTotal = playerOneTotal + playerScores[i];
                playerTwoTotal = playerTwoTotal + playerScores[i+1];
                playerThreeTotal = playerThreeTotal + playerScores[i+2];
                holeNum++;

        System.out.printf("TOTAL \t%d \t", courseParTotal);
        System.out.printf("%d (%d) \t", playerOneTotal, playerOneTotal-courseParTotal);
        System.out.printf("%d (%d) \t", playerTwoTotal, playerTwoTotal-courseParTotal);
        System.out.printf("%d (%d)\n", playerThreeTotal, playerThreeTotal-courseParTotal);
```

Winner()

```
        if(playerOneTotal < playerTwo Total && playerOneTotal < playerThreeTotal)
                winner = 1;
                System.out.println(“Player %d wins!”, winner);
        etc.

        if(playerOneTotal == playerTwoTotal && playerOneTotal < playerThreeTotal)
                System.out.println(“Player 1 and 2 are tied for the win!”);
        etc.
```

3. Looking Back / Self Reflection

My first major bug was getting the program to properly read in the player scores and then convert them to an array of integers. I struggled for a few hours trying various solutions, including use of .trim() and various methods of converting Strings and chars to ints. I thought maybe there was an issue with java’s internal casting of Int type to the (primitive) int type. In the end I got some help from TA Reed who suggested my issue was that my Scanner object initially was reading in the first char as the \n from the first line of the file where the number of players is read in. This was causing the code to constantly give an error because java is unable to convert a \n to an int. He showed me how I just needed to add a line of code to advance the Scanner to the next of the text file where the player score integers began.

After getting everything working I went back and added some extra credit features. Instead of requiring the user to name their input files course.txt and scores.txt my code now just takes in two command arguments and passes them into the other methods called in main. I added a FileOut method that writes the table of results to a text file. Here I struggled getting the formatting right. I found out that for file output in Windows you must use \r\n instead of just \n for a new line. Otherwise in Windows your entire output is on a single line.

I did a few other quick bonus features – the number of holes can be arbitrarily set or can even be randomly generated. Instead of the user having to create their own input files my code now will randomly generate a course with new pars and player scores each time.

I wanted to go back and make the number of players arbitrary or even random, instead of defined as 3, however it would take a lot of editing of my code as I did not plan from the start the idea of using a different number of players. For example, my code uses three variables called playerOneScore, playerTwoScore, playerThreeScore, to hold the totals for each player – instead I should have had an array of int with length equal to number of players. So that’s the biggest thing I learned in this assignment – plan ahead for possible future features when starting code from scratch!