# COP3022 Programming Project # 3

## Grading Sheet

**Total Score:** (100 points total): \_\_\_\_\_\_\_\_\_ Name: \_\_\_*Kyle Ligon \_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_

**NOTE all items on grade sheet refer to correctly implementation of the item.**

1. **General issues:**
   1. \_x\_\_\_\_\_\_ (3 pts) Well-documented
   2. \_\_\_\_\_\_\_ (-5 pts) Javadoc HTML file (correct format and tags)
   3. \_\_\_\_\_\_\_ (-5 pts) Good programming style (formatting, variable names, no magic numbers etc)
   4. \_\_\_\_\_\_\_ (-5 pts) Incorrect program structure (5 separate class, and driver program)
   5. \_\_\_\_\_\_\_ Compiles Y/N \_\_\_\_\_\_\_\_\_\_\_\_(-70 points)
2. **Player Class:**
   1. \_\_x\_\_\_\_\_ (3 pts) Instance variables
   2. \_\_x\_\_\_\_\_ (3 pts) Static Variable to handling ID Number
   3. Methods
      1. \_x\_\_\_\_\_\_ (3 pts) Constructors as specified.
      2. \_x\_\_\_\_\_\_ (3 pts) toString - returns a neatly formated String representing the player's information and all their scores.
      3. \_x\_\_\_\_\_\_ (3 pts) calculateHandicap - abstract method that returns a double representing the players handicap
3. **Golfer class**
   1. \_\_\_\_\_\_\_ (-5 pts) Does not work as required in Programming Project 2
   2. \_\_x\_\_\_\_\_ (7 pts) Calculate Handicap method – matches abstract method of superclass
   3. \_\_x\_\_\_\_\_ (6 pts) Methods modified to handle change in Score class
4. **Score Class**
   1. \_\_\_\_\_\_ (-5 pts) Does not work as required in Programming Project 1
   2. \_\_\_x\_\_\_ (3 pts) Course information now object instance variable
   3. \_\_\_x\_\_\_ (5 pts) Methods modified to handle Course object instance variable
5. **Course Class**
   1. \_\_\_x\_\_\_ (2 pts) Correct instance variables
   2. \_\_\_x\_\_\_ (2 pts) Constructors – Default and Parameterized
   3. \_\_\_x\_\_\_ (2 pts) Access and Mutator methods
   4. \_\_\_x\_\_\_ (2 pts) toString Method
6. **Bowler class**
   1. \_\_\_x\_\_\_\_ (4 pts) Instance variables
   2. \_\_\_x\_\_\_\_ (4 pts) Constructor (Parameterized and Default)
   3. \_\_\_x\_\_\_\_ (4 pts)  IDNum set programmatically in constructor and mutator
   4. \_\_\_x\_\_\_\_ (2 pts) Access and mutator methods
   5. \_\_\_x\_\_\_\_ (7 pts) Calculate Handicap method
   6. \_\_\_x\_\_\_\_ (3 pts) Add Score method
7. **BowlerScore** **class** 
   1. \_\_\_x\_\_\_\_ (2 pts) Instance variables
   2. \_\_\_x\_\_\_\_ (2 pts) Constructor (Parameterized and Default)
   3. \_\_\_x\_\_\_\_ (2 pts) Access and Mutator methods
   4. \_\_\_x\_\_\_\_ (2 pts) toString Method
8. **FieldOutOfBounds Exception** 
   1. \_\_\_x\_\_\_\_\_ (5 pts) Exception class properly written to display message.
   2. \_\_\_x\_\_\_\_\_ (5 pts)Throw in correct place in code to catch any errors.
9. **PlayersTester class** 
   1. \_\_\_x\_\_\_\_ (4 pts) Tests all the methods in each class directly or indirectly including error conditions
   2. \_\_\_x\_\_\_\_ (4 pts) Reads Player’s (Golfer and Bowler) information and a set of scores from a file via a command line redirection using bat file format provided.  Data should additionally test error conditions
   3. \_\_\_x\_\_\_\_ (3 pts) Creates an ArrayList of Players.
   4. \_\_\_x\_\_\_\_ (3 pts) Populates the ArrayList with both Golfer and Bowler objects.
   5. \_\_\_x\_\_\_\_ (4 pts) Call the toString method on each object in the Arraylist polymorphically
10. **UML Diagram**
    1. \_\_\_x\_\_\_\_ (5 pts) Class diagram (with all methods, instance fields, associations)
11. **Other Issues**
    1. \_\_\_\_\_\_\_ (-5 pts) File data not in correct format.
    2. \_\_\_\_\_\_\_ (-5 pts) Failed to use mutator methods to set instance variables.
    3. \_\_\_x\_\_\_\_ (+ 5 pts) Self graded grade sheet turned in.
    4. \_\_\_\_\_\_\_ (-5 pts) Code ensures all instance field are valid, error code in both constructor and methods.  Error messages provided as needed.  Program continues after error.
    5. \_\_\_x\_\_\_\_ (submission,  etc including input.txt and .bat file )

Comments: