CSC20 Programming Project 2

Note: make sure that your methods accept parameters of type object where applicable

Use instanceof to check the data type of the object

Use type casting to cast the object into the appropriate data type if needed

**Create the following hierarchy of classes**

1. Create a class called **Player** with the following UML diagram

|  |
| --- |
| Player |
| -String name  -double salary  -int number |
| +Player(String name, double salary, int number)  setter methods  getter methods  equals method  tostring method |

1. Create a class called **BasketBallPlayer** that extends the player class.

|  |
| --- |
| BasketBallPlayer |
| -int threePointer  -int turnOvers  -int assist  -int rebounds  -int twoPointers  ~~-String teamName~~ |
| +constructor  +setter methods  +getter methods  +equals  +toString  +getTotalPoints(): int// returns the total points made by the player |
|  |

***Starter Code for BasketBall Player***

public class BasketballPlayer extends Player implements Comparable{

private int threePointer;

private int turnover;

private int assist;

private int rebound;

private int twoPointer;

private String team;

public BasketballPlayer(String name, double salary, int number, int three, int turnover, int assist, int rebound, int two, String team) ;

1. Create a class called **SoccerPlayer** that extends the player class.

|  |
| --- |
| Soccer |
| -int goals  -int fouls  -int redCard  -int yellowCard  -Int offside  -int assist  ~~-String teamName~~ |
| +constructor  +getter methods  +setter methods  +toString method  +equalsMethod// accepts an object as its parameter  +pullOutPlayer(): Boolean// if the player has more than three fouls or two yellow card then the player cannot play and this method should return false |

***Starter Code for Soccer Player***:

public class SoccerPlayer extends Player implements Comparable{

private int goals;

private int fouls;

private int red;

private int yellow;

private int offside;

private int assist;

private String team;

public SoccerPlayer (String name, double salary, int number, int goals, int fouls, int red, int yellow, int offside, int assist, String team)

1. Create a class called **Team**

|  |  |
| --- | --- |
| Team |  |
| String TeamName  ArrayList<Player> team |  |
| +constructor(teamName, team) |  |
| +getter method | Returns the object team of ArrayList type |
| +add(Object o) | Adds the object to array list |
| +add(Object o, int pos) | Adds the object at the given position in arraylist |
| +remove(int index) | Removes the object(Player) from the list at index |
| +equals(Object other) | Returns true if the team is the same as the team other. Need a loop to compare all the players. |
| +toString | Returns a string representing all the players in the team. A for loop is needed |

1. Create a class called **SoccerTeam** extends **Team**. This class has an arraylist of 15 players

|  |  |
| --- | --- |
| +constructor() | Reads the info for the players from a file and creates an object of SoccerPlayer and then stores it in the arrayList called team. Then call super constructor and pass team name and array list. |
| +add(Object o)  +add(Object o, int pos) | Checks that object is an instance of SoccerPlayer and calls add method of superclass.  Adds the object at the given position |
| +selectTheBest() | Return the player with the most goals |
| +selectMostRed | Returns the player with the most red card |

1. Create a class called **BasketBallTeam** extends **Team** that has the same/similar features as step 4
2. Create a driver class that does the following:
   1. Create a main method
      1. Create two data files for the basketball team and the soccer team. First line should give the Team name. These two files will be used by the constructors in the Soccer class and Basketball class.
      2. Create an object of SoccerTeam call it bearCat
      3. Create an object of BasketBallTeam call it kings
      4. Call the methods in step 6 (b)
   2. Create the following methods
      1. Create a method called listPlayers, this methods accepts an arraylist of players and displays the info about all the players. Call this method two times using the object kings and bearCat. You must use the same method
      2. Write a method that accepts an array of SoccerPlayer and lists all the players with fouls.
      3. Write a method called mostRebound that displays the player with the most rebound
      4. Write a method called searchPlayer that accepts an arraylist of type Player and a name of a player. This method searches the list, if the player is found display the info, otherwise print a message that the player does not exist.
      5. Write a method that display the best player with the most goals/points
      6. Write a method that takes the team name as parameter and simulates series of 10 games and each game is a random play. For each game, you need to generate a random number, if the random number is bigger than 0.5.its a win; otherwise the team loses a game. This method prints the number of wins and losses of each team. Call this method twice, once by passing in “Bear Cat” as team name and other for passing in “Kings” as the team name from the main method.
      7. Write a method that removes a player with most red cards(Soccer) / least points(Basketball)and adds a new player in his place.

**Programming Guidance**

1. Note: make sure that your methods accept parameters of type ***Object***  where applicable
2. Use ***instanceof*** to check the data type of the object
3. Use type casting to cast the *Object* into the appropriate data type if needed
4. The *equals* method for each class only compares the names of the players

**Grading policies**

1. Comments/ JavaDoc is necessary for each class
2. Indentation and good programming style
3. Data validation/reading from a file using *try catch* exception handling
4. Follow the assignment specifications as given