Practice Mini Prelim

PECANWOOD COLLEGE

Prepared for Life

INFORMATION TECHNOLOGY MINI PRELIM EXAM – PRACTICAL GRADE 12

NAME: _____ **GRADE:** _____

DATE: **MARKS: 125 MARKS**

MODERATOR: MR R NAINAR **TIME: 180 MINUTES**

EXAMINER: MR SC EILERTSEN

INSTRUCTIONS:

1. This test is made up of 9 pages. Please ensure that your paper is complete.
2. You will be provided with
   a. A digital text file for question one.
   b. A database for question two that you can open and run in Ms Access.
   c. A digital SQL answer sheet that you open in Ms Word
3. You may use a non-programmable calculator where relevant.
4. Compile and run your code often. This makes troubleshooting a lot easier.
5. When your program is complete you must send the source code (all classes) to the printer with a multi-line comment that includes your name, grade, terminal number and date.
6. You must also print your SQL answer sheet for question 2 with your name, grade, terminal number and date.
7. Credit is given for writing style, ease of reading, user friendliness, indentation, use of uppercase and lowercase letters and logic that is efficient and re-useable.

Section One **Reading from a text file into an array of objects**

1.1) The text file "players.txt" contains the details of a football club i.e.
Full name, ID number, date of birth, professional (yes or no), professional number (if relevant), position, team.
This project is a work in progress and therefore not all the data is available at this time.

Create an array of player-objects by reading in the text file "players.txt" using the Scanner class.

To do this you must create three classes (the UI class with a small main method, a manager class with all the methods, and a player-class from which to instantiate player-objects.

- Call your class with the main method **"PlayerUI"** (3)
- Call your manager class **"ManagePlayer"** (22)
- Call your player class **"Player"** (10)

Total 35

Class: PlayerUI

This will create the manager object and call its methods – nothing else.

Class: ManagePlayer

To start off with the ManagePlayer class will only have a constructor method that will read in the text file. As it does so it will separate the first name from the last name, and correct the professional status field (see below *).

The text file has the following fields, in this order . . .

Full name, ID number, date of birth, professional (yes or no), professional number (if relevant), position, team.

When you open your text file it will look like this . . .

```
Carala Goodbody#7106233650088#23/06/1971#180#no####
Rudibert Roperse#7206135574080#13/06/1972#179#no####
Holman Bolger#7103047056195#04/03/1971#183#yes####
Cosimo Zarager#7107196900992#19/07/1971#173#yes####
```

This soccer club project is a work in progress and therefore not all the data is available; this is why some of the fields on the right hand side are still blank.

* Note the problem with field five (above) - the pro status has been incorrectly captured as "yes/no" – this must be converted to "true/false" in your program.

Class: Player

Here is the **class diagram** that must be used to create each of the player objects.

Player Class	
Fields	**Description**
- String firstName	Holds the players first name
- String lastName	Holds the players last name
- String id	Holds the player's South African ID number starting with the date of birth – YY MM DD
- date of birth	Day, month and year.
- int height	Holds the height of the player in centimetres
- boolean professional	Professional or not. True or False This field is meant to contain "true" or "false" and not "yes" or "no". Your code in your manager class must read in the yes/no values and then change them to true/false before transferring them to the player class.
- String pronumber	If the player is professional, this field will hold their professional league reference number. This field is currently empty.
- String position	The position the player plays. This field is currently empty.
- String team	The team that the player plays in (A, B, C etc) This team is currently empty
Methods	
+ toString	Joins **selected** fields together into a single String i.e. ID, lastname, initial of first name only, DOB, height, professional status, **in this order.**

1.2) Now add another method to your class **ManagePlayer**. It must iterate through your player array and print out all the records using the Player toString method. PlayerUI must call this method.

Here is a copy of the final output that prints all the player-objects – yours must look very similar.

NOTE:

- The players are listed from one, not from zero.
- The field for professional is boolean.
- The empty fields have been left out of the toString method .
- The name displays the first name initial only.

```
Player   ID             Full Name      DOB    Height   ProStatus
=================================================================
Player 1 7106233650088  Goodbody C   23/06/1971  180   false
Player 2 7206135574080  Roperse  R   13/06/1972  179   false
Player 3 7103047056195  Bolger   H   04/03/1971  183   true
Player 4 7107196900992  Zarager  C   19/07/1971  173   true
```

(12)

1.3) Now add getter and setter methods (accessor and mutater methods) to allow missing proNumber to be captured from the keyboard for each of the professional players and then stored into the relevant array object. Pay attention to a useful prompt so that the user knows who the current player is.

(19)

Methods	
+ toString	Joins **selected** fields together into a single String i.e. ID, lastName, initial of first name only, DOB, height, professional status, **in this order.**
+ toString2	Joins **selected** fields together into a single String i.e.ID, lastName, initial of first name only, professional status and profession reference number, **in this order.**
+ setProNumber	If a player is professional, and if their professional reference number is missing, this method will allow the missing number to be captured from the keyboard and stored into the relevant player object.

1.4) Once the professional players have their ProNumber create a new method to print out the players name with the newly captured ProNumber

The output will should look like this shown below. Player 1 and 2 are not professionals and therefore ProNumber is left blank. Player 3 and 4 are professionals and therefore their professional reference Numbers were entered from the keyboard. This output will require a different toString method as the selected fields are not the same as the first toString method. (9)

```
Player   ID              Full Name    Pro   ProNumber
============================================================
Player 1 7106233650088   Goodbody C   false
Player 2 7206135574080   Roperse  R   false
Player 3 7103047056195   Bolger   H   true  KA34548-56
Player 4 7107196900992   Zarager  C   true  ZA67549-43
```

Send your three classes to the printer with your name, grade, terminal number and date in a multi-line comment

Section Two: (75)

Grade 12 Mini Prelim 2020. Memo.

```
1 // Player UI class. Grade 12 Mini Prelim April 2020
2
3 public class PlayerUI
4 {
5    public static void main (String[]args)
6    {
7       // Ques 1.1
8       ManagePlayer myManagePlayer = new ManagePlayer();
9
10      // Quest 1.2
11      myManagePlayer.printArr();
12
13      // Ques 1.3
14      myManagePlayer.setProNumber();
15
16      // Ques 1.4
17      myManagePlayer.printPro();
18
19
20   } // end main
21 } // end class
```

1.1

1.2

1.3

1.4

(A)

```
1 // Manage Player class. Mini prelim 2020
2
3 import java.io.*;
4 import javax.swing.JOptionPane;
5 import java.util.Scanner;
6
7 public class ManagePlayer
8 {
9
10    static int size = 0; // global variable
11    static Player[] myPlayers = new Player[20]; // global variable
12
13    // Constructor method
14    public ManagePlayer()
15    {
16       boolean professional; // local variable
17       try catch
18       {
19          Scanner scFile = new Scanner(new File("players.txt"));
20          while (scFile.hasNext())
21          {
22             String line = scFile.nextLine();
23             Scanner scLine = new Scanner(line).useDelimiter("#");
24             String name = scLine.next();
25
26             // separate the first name from the last name.
27             int space = name.indexOf(" ");
28             String firstName = name.substring(0, space);
29             String lastName = name.substring(space + 1);
30
31             String id = scLine.next();
32             String dob = scLine.next();
33             int height = scLine.nextInt();
34
35             // deal with the yes/no - true/false problem from the text file
36             professional = false;
37             String professionalSt = scLine.next();
38
39             if (professionalSt.equals("no"))
40                professional = false;
41             else if (professionalSt.equals("yes"))
42                professional = true;
43             else
44             {
45                System.out.println("Error with professional");
46                System.exit(0);
47             }
48
49             String proNumber = scLine.next();
50             String position = scLine.next();
51             String team = scLine.next();
52
53             // Create new player object - pass parmeters to constructor
54             myPlayers[size] = new Player(firstName, lastName, id, dob,height, professional, proNumber, positi
on, team);
55             size++;
56
57          } // end while
58       } // end try
59
60       catch (FileNotFoundException ex)
61       {
62          JOptionPane.showMessageDialog(null, "Error file not found.");
63       } // end catch
64
65    } // end constructor method
66
```

Q 1.1 (22)

Name separation

yes/no → true false

empty fields

(B)

```
67    } // end constructor method
68
69    // Ques 1.2
70    // Print whole array, selected fields
71     public void printArr()
72     {
73       String tab = "\t";
74
75       System.out.println("Player" + tab + "ID" + tab + tab + tab + tab + tab +   "Full Name" + tab + tab +
"DOB" + tab + "Height" + tab + "ProStatus");
76       System.out.println("=================================================================");
77       for (int i = 0; i < size; i++)
78            System.out.println("Player " + (i+1) + "\t" + myPlayers[i].toString());
79
80       System.out.println();
81     } // end printArr
82
83     // Ques 1.3
84     // If pro, set the pronumber if missing
85     public void setProNumber()
86     {
87       for (int i = 0; i < size; i++)
88       {
89          String theID = myPlayers[i].getIDNumber();
90          boolean isProfessional = myPlayers[i].getProfessional();
91          String hasProNumber = myPlayers[i].getProNumber();
92
93          if((hasProNumber.length() == 0) && (isProfessional == true)) // professional but no number
94          {
95             String proNumber = JOptionPane.showInputDialog("Enter missing professional number for player ID
" + theID);
96             myPlayers[i].setProNumber(proNumber);
97          }
98       }
99
100    } // end setProNumber
101
102    // Ques 1.4
103    // Print the array of player objects with their pro status and pronumber
104    public void printPro()
105    {
106      String tab = "\t";
107
108      System.out.println("Player" + tab + "ID" + tab + tab + tab + tab + tab +   "Full Name" + tab + "Pro" +
 "\t" + "ProNumber");
109      System.out.println("=============================================================");
110
111      for (int i = 0; i < size; i++)
112           System.out.println("Player " + (i+1) + "\t" + myPlayers[i].toString2());
113
114      System.out.println();
115    } // end printPro
116
117 } // end class
```

Q 1.2 — (10)

Field order

Q 1.3 — (12)

Q 1.4 — (5)

```java
// Template / Object class from which to create player objects.
// Mini prelim 2020

public class Player
{

    private String firstName, lastName, id, dob, proNumber, position, team;
    private int height;
    private boolean professional;

    // Constructor method
    public Player(String fn, String ln, String i, String db, int h, boolean pr, String pn, String po, String t
)
    {
        firstName = fn;
        lastName = ln;
        id = i;
        dob = db;
        height = h;
        professional = pr;
        proNumber = pn;
        position = po;
        team = t;
    } // end constructor


    // Ques 1.2
    // Method prints selected data
    public String toString()
    {
        return id + "\t" + lastName + "\t" + firstName.charAt(0) + "\t" + dob + "\t" + height + "\t" + professi
onal;
    }

    // Ques 1.3
    public boolean getProfessional()
    {
        return professional;
    }

    // Ques 1.3
    public String getProNumber()
    {
        return proNumber;
    }

    // Ques 1.3
    public String getIDNumber()
    {
        return id;
    }

    // Ques 1.3
    public void setProNumber(String pn)
    {
        proNumber = pn;
    }

    // Ques 1.4
    // Method prints selected data, including professional status and professional number
    public String toString2()
    {
        return id + "\t" + lastName + "\t" + firstName.charAt(0) + "\t" + professional + "\t" + proNumber;
    }
}
```

match

Q 1.1 — 10

initial

Q 1.3 — 6

Q 1.4 — 3

D