INFORMATION TECHNOLOGY – PAPER I

GRADE 12

AUGUST / SEPTEMBER 2022

Time: 3 hours **Total: 150 Marks**

EXAMINERS: E van Aarde (Section A), R Viljoen (Section B)
MODERATORS: S Barber, M Ounaceur, M Walker

Memorandum

SECTION A STRUCTURED QUERY LANGUAGE

QUESTION 1

1.1	SELECT * ✓ FROM tblRiders ✓ WHERE YoungRider = TRUE ✓ ORDER BY Age ✓	**4**
1.2	SELECT * FROM tblRiders ✓ WHERE Position IS NOT NULL ✓ AND ✓ Age > 30 ✓ ORDER BY Age DESC ✓ **Accept:** WHERE Position > 0 WHERE Position <> 0	**5**
1.3	SELECT CountryName ✓, COUNT(*) ✓AS [Number of Riders] ✓ FROM tblRiders, tblCountries ✓ WHERE tblRiders.CountryID = tblCountries.CountryID ✓ GROUP BY CountryName ✓ HAVING COUNT(*) > 5 ✓ **Accept:** INNER JOIN	**7**

1.4	SELECT RiderName, RiderSurname, Age, Position FROM tblRiders ✓ WHERE Age ✓ = (SELECT MIN(Age) ✓ FROM tblRiders) ✓	**4**
1.5	SELECT TOP 10 ✓ RiderName, RiderSurname, Position ✓ FROM tblRiders WHERE Position > 0 ✓ ORDER BY Position ✓	**4**
1.6	SELECT RiderName, RiderSurname, CountryName, Colour ✓ FROM tblRiders, tblCountries, tblJerseys ✓✓ WHERE tblRiders.CountryID = tblCountries.CountryID ✓ AND ✓ tblRiders.RiderID = tblJerseys.RiderID ✓ **Accept:** INNER JOIN	**6**
1.7	SELECT CountryName ✓, LEFT(CountryName, 3) ✓ & ✓ INT ✓ (RND(CountryID) ✓* 100 + 1 ✓) AS Code, TotalRiders - RidersInCompetition ✓ AS [Riders Not In Competition] ✓ FROM tblCountries WHERE TotalRiders - RidersInCompetition > 0 ✓	**9**
1.8	INSERT INTO tblRiders ✓ (RiderName, RiderSurname, Age, Position, YoungRider, CountryID) ✓ SELECT "Sebastian", "Schonberger" ✓, Age, 34, YoungRider, CountryID ✓ FROM tblRiders WHERE RiderID = 45 ✓	**5**
1.9	UPDATE tblRiders ✓ SET ✓ Position = 120 ✓ WHERE RiderID = 46 ✓	**4**
1.10	DELETE FROM tblRiders ✓ WHERE RiderID = 49 ✓	**2**

50 marks

SECTION B OBJECT ORIENTED PROGRAMMING

QUESTION 2

```
import java.time.Duration;
import java.time.LocalTime;

/**
 *
 * @author rickusv
 */
//Question 2 [14]
//Question 6.1 [10]
//Question 2.1 (1)
public class YellowRider
{ //1 class name

    //Question 2.5 (2)
    public static final String JERSEY_COLOUR = "[ YELLOW JERSEY ]"; //2

    //Question 2.2 (3)
    private String name, team, overall;
    private int points;
    //1 private, 1 named correctly, 1 typed correctly

    //Question 2.3 (3)
    public YellowRider(String n, String t, String o)
    { //1 header with correctly named params
        this.name = n;
        this.team = t;
        this.overall = o;//1 assign param values to fields
            points = 0; //1 set default
    }

    //Question 2.4 (2)
    //2    -1 per mistake
    public String getName()
    {
        return name;
    }
```

```
public String getTeam()
{
    return team;
}

public String getOverall()
{
    return overall;
}

    //Question 2.6 (2)
@Override
public String toString()
{ //1 Method header
    return name + "\t" + team + "\t" + overall; //1 return correct fields correct format
}

//Question 2.7 (1)
public String fileLine()
{ //
    String temp = name + "#" + team + "#" + overall; //1 correct fields

    return temp;
}

//*********************************************************************
//Question 6.1 (10)
public void setOverall(LocalTime start, LocalTime end) { //1 params
    Duration diff = Duration.between(start, end); //1 determine difference
    long sec = diff.getSeconds();  //JDK 8
    //long sec = diff.toSeconds();  //1 convert to seconds elapsed

    int hh = Integer.parseInt(overall.substring(0, 2)) + LocalTime.ofSecondOfDay(sec).getHour();
    int mm = Integer.parseInt(overall.substring(3, 5)) + LocalTime.ofSecondOfDay(sec).getMinute();
    int ss = Integer.parseInt(overall.substring(6)) + LocalTime.ofSecondOfDay(sec).getSecond();
                //3 get hh, mm and sec from elapsed time and add to each of overall
    if (ss >= 60) {
        mm++;
        ss -= 60; //1 add seconds overflow to minutes
    }
    if (mm >= 60 ) {
        hh++;
        mm -= 60; //1 add minutes overflow to hours
```

```
        }
        String strSS = ss + "";
        String strMM = mm + "";
        if (ss < 10) {
            strSS = "0" + ss;
        }
        if (mm < 10)
            strMM = "0" + mm;   //1 add leading 0

        overall = hh + ":" + strMM + "," + strSS; //1 set new time in correct format

    }

}
```

QUESTION 3

```
Question 3 [13]

Question 3.1 (2)
public class GreenRider extends YellowRider { //1 class name //1 extends

Question 3.2 (2)
    private int points; //1 private //1 field

Question 3.3 (1)
    public static final String JERSEY_COLOUR = "[ GREEN JERSEY ]"; //1

Question 3.4 (3)
    public GreenRider(String n, String t, String o, int p) { //1
        super(n, t, o); //1
        points + p; //1
    }

Question 3.5 (2)
    public int getPoints() { //1
        return points;
    }

    public void setPoints(int points) {//1
        this.points += points;
    }

Question 3.6 (2)
    @Override
    public String fileLine() {//1 override //1 return
        return super.fileLine() + "#" + points;
    }

Question 3.7 (1)
    @Override
    public String toString() { //1
        return super.toString() + "\t" + points;
    }
}
```

QUESTION 4

```
public class Controller {//Q 4.1  //new class 1

    //Question 4.2 (4)
    private YellowRider[] arr = new YellowRider[200];   //1 private //1 YellowRider arr 200
    private int count = 0;                  //1 counter //1 counter =0

    //Question 4.3 (12)
    public Controller(String fileName) //1 Method header with filename
    {
        try //1 if file exists
        {
            Scanner scFile = new Scanner(new File(fileName)); //1 open file

            while (scFile.hasNext()) //1 loop
            {
                String[] line = scFile.nextLine().split("#");   //1 read line
                String temp = "";                               //1 tokenise

                //1 test line
                if (line.length == 3) {                         //1 add to array
                    arr[count] = new YellowRider(line[0], line[1], line[2]); //1 create object
                } else {            //1  correct constructor
                    arr[count] = new GreenRider(line[0], line[1], line[2], Integer.parseInt(line[3]));
                }

                count++;    //1 increment counter
            }

        } catch (FileNotFoundException ex) {
            System.out.println("File not found");  //1 display message
        }
    }

    //Question 4.4 (5)
    public String listAll() {

        String temp = listJerseyRiders();  //1 initiate string
                                           //listJerseyRiders can be called here or UI

        for (int i = 0; i < count; i++) {//1 loop
            temp += i + 1 + "\t" + arr[i].toString() + "\n"; //1 rider pos
```

```
    }                           //1 return + object.toString
    return temp;                //1 new line
}

//Question 4.6 (6)
public GreenRider findGreenRider (){ //1 method header
    int lead = 0;
    GreenRider temp = null;

    for (int i = 1; i < count; i++) {
        if (arr[i] instanceof GreenRider) { //1
            //1 type cast
            if (lead < ((GreenRider) arr[i]).getPoints()) //1 compare
            {
                lead = ((GreenRider) arr[i]).getPoints(); //1 new points leader
                temp = (GreenRider) arr[i];
            }
        }
    }
    return temp;//1 return object
}

//Question 4.7 (5)
public String listJerseyRiders(){
    String temp = "\t" + YellowRider.JERSEY_COLOUR + " ]\n"; //1;
    temp += arr[0].toString() + "\n"; //1
    temp = temp + "\t" + GreenRider.JERSEY_COLOUR + " ]\n"; //1
    temp = temp + findGreenRider().toString() + "\n\n";  //1 add heading and rider
                  //1 method call
    return temp;
}

//Question 4.5 (6)
public void sort() {
    for (int i = 0; i < count - 1; i++) //1 both counters
    {
        for (int j = i + 1; j < count; j++) //1 both counters
        {
            if (arr[i].getOverall().compareTo(arr[j].getOverall()) > 0) {  //1 compare //1 object method call
                YellowRider temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;              //2 swop
            }
```

```
            }
        }
    }

    //Question 6.2 (17)
    public void stageResults(String fn) {
        LocalTime start = LocalTime.of(12, 35, 0);//1
        try {
            Scanner scFile = new Scanner(new File(fn));

            while (scFile.hasNext()) //marks for file reading already assigned
            {
                String[] line = scFile.nextLine().split("#"); //1 read and tokenise
                int i = 0;
                boolean found = false;
                while (!found || i < count) //2 flagged search to match object
                {
                    if (line[0].equals(arr[i].getName())) //1 match object
                    {
                        LocalTime stageTime = LocalTime.parse(line[2], DateTimeFormatter.ofPattern("HH:mm,ss"));//2 read
and parse
                        arr[i].setOverall(start, stageTime); //1 method call

                        if (line.length == 4) { //1
                            if (!(arr[i] instanceof GreenRider)) { //1
                                arr[i] = new GreenRider(arr[i].getName(), arr[i].getTeam(), arr[i].getOverall(),
Integer.parseInt(line[3]));
                                            //1
                            }
                            ((GreenRider) arr[i]).setPoints(Integer.parseInt(line[3])); //2 set points
                        }
                        found = true;
                    }
                    i++;
                }

            }
            sort();                                 //1 sort
            PrintWriter p = new PrintWriter(new File("src/resources/overallstage11.txt"));  //1 printwriter

            for (int i = 0; i < count; i++) {
                p.println(arr[i].fileLine());   //1 write to file
            }
```

```
            p.close();                          //1 save

        } catch (FileNotFoundException ex) {
            System.out.println("File not found");
        }

    }
    //**************************************************************************
}
```

---

100 marks

Total: 150 marks