*Place candidate's barcoded sticker here.*

*This sheet must be stapled to the front of each candidates' submission.*

# I.T. PRACTICAL EXAMINATION 2015

| | | | |
|---|---|---|---|
| **Version:** | **FINAL Marksheet (October 2015)** | **Prog. Lang.** | |
| **Candidate:** | **Test** | | |
| **Marker:** | **D Gruijters** | **120** | **0** |
| **Checksum:** | 0000-0000-0120 | **Total** | **Pupil** |

| | | | |
|---|---|---|---|
| | **SQL** | | |
| 1.1 | SELECT Country, FullName ✓ FROM tblPlayers ✓<br>ORDER BY Country, Fullname ✓ | **3** | |
| 1.2 | SELECT * ✓ FROM tblFinals ✓ WHERE FinalYear < 2000 ✓ | **3** | |
| 1.3 | DELETE * ✓ FROM tblPlayers WHERE PlayerID = 101 ✓<br>Do not accept PlayerName = 'Bob Rogers' as a criterion. | **2** | |
| 1.4 | UPDATE tblTournaments ✓ SET PrizeMoneyZAR ✓ = PrizeMoney * 11.51 ✓ | **3** | |
| 1.5 | SELECT MONTH(DateOfBirth) AS BirthMonth ✓, COUNT(*) AS NumBirthdays ✓<br>FROM tblPlayers GROUP BY ✓ MONTH(DateOfBirth) ✓<br>(-1 mark if no aliases are specified at all)(MYSQL: Accept GROUP BY BirthMonth)(COUNT(Field) also acceptable) | **4** | |
| 1.6 | SELECT Country ✓, COUNT(*) ✓ AS TotalPlayers FROM tblPlayers<br>GROUP BY Country ✓<br>HAVING ✓ COUNT(*) > 10 ✓<br>ORDER BY COUNT(*) DESC ✓<br>(Accept ORDER BY Country DESC as well) (MYSQL: Accept HAVING TotalPlayers, ORDER BY TotalPlayers) | **6** | |
| 1.7 | SELECT TournamentName ✓ FROM tblTournaments<br>WHERE PrizeMoney ✓ > (SELECT AVG(PrizeMoney) ✓ FROM tblTournaments) ✓ | **4** | |
| 1.8 | SELECT FullName, TournamentName, FinalYear ✓<br>FROM tblFinals, tblPlayers, tblTournaments ✓✓<br>WHERE tblFinals.TournamentID = tblTournaments.TournamentID ✓<br>AND tblFinals.WinnerID = tblPlayers.PlayerID ✓<br>AND FinalYear MOD 4 = 0 ✓ AND FinalYear MOD 400 <> 0 ✓ AND FinalYear > 1995 ✓<br>(Accept any solution that works for future years past 2015)(Accept any equivalent of the MOD operator)<br>(Accept INNER JOIN instead of WHERE tblFinals.TournamentID = tblTournaments.TournamentID) | **8** | |
| 1.9 | INSERT INTO tblFinals ✓ (TournamentID, FinalYear, FinalName, WinnerID, RunnerUpID) ✓<br>SELECT ✓ 2 ✓, FinalYear, FinalName, WinnerID, RunnerUpID ✓<br>FROM tblFinals WHERE TournamentID = 1 ✓ AND FinalYear = 2015 ✓<br>(Penalise 1 mark for incorrect syntax such as added brackets or the word "VALUES") (Not necessary to list fields but make sure SELECT statement fields matches order of tblFinals fields) | **7** | |
| | **Programming** | | |
| 2.1 | **Player Class:** Class header is correct ✓ | **1** | |
| 2.2 | **Fields/Properties:** attributes declared private ✓ name correctly ✓ and correct type ✓ (-1 Penalise for each class where non-private attributes are used as it's a fundamental error and misunderstanding of OOP) | **3** | |
| 2.3 | **Constructor:** header is correct ✓ with correct parameters ✓ and values assigned to correct parameters ✓(Names of parameters must match names given in the question. Penalise in 2.3 OR 3.3) | **3** | |
| 2.4 | **getCountryCode:** method header is correct ✓ check for a space in country ✓ generate code correctly for countries with spaces ✓ generate code correctly for countries without spaces ✓ (Deduct 1 mark if not uppercase) | **4** | |
| 2.5 | toString: method header correct ✓ getCountryCode called ✓ and concatenated in correct format with fullname and returned ✓ | **3** | |

| | | | |
|---|---|---|---|
| 3.1 | **Match Clas**s: Class header is correct ✓ | **1** | |
| 3.2 | **Attributes:** matchCode and score declared correctly ✓ player1 and player2, BOTH attributes declared as Player objects ✓✓(-1 again for non-private attributes as it's a fundamental error and misunderstanding of OOP) | **3** | |
| 3.3 | **Constructor:** MatchCode parameter is of the correct type ✓ player1 and player2 attributes are Player objects ✓ Values assigned to correct attributes ✓ and score initialised to "X" ✓(Names of parameters must match names given in the question. Penalise in 2.3 OR 3.3) (-1 if Score parameter is present) | **4** | |
| 3.4 | **getMatchCode:** accessor method header is correct and returns the matchCode correctly ✓ | **1** | |
| 3.5 | **setScore:** Method header, including parameter, is correct and correct assignment ✓ | **1** | |
| 3.6 | **toString:** method header correct ✓ both player objects' toString method concatenated ✓ with the 'vs.' concatenated ✓ Check if there is a score ✓ and concatenate accordingly ✓ or return not played ✓(Accept matchCode if included but do not award mark) (Do not penalise for inefficient code) | **6** | |
| | | | |
| 4.1 | **TournamentManager:** Class header is correct ✓ | **1** | |
| 4.2 | **PROPERTIES:** Private✓ array of player objects ✓ and array match objects declared correctly✓ and both of correct size (32 and 16 respectively)✓(Do not penalise or award marks for counter variables here or anywhere) | **4** | |
| 4.3 | **Constructor:** Accepts filename as a parameter ✓ Display error if file does not exist ✓ Open file for reading ✓ Loop through every line in the file (while or for loop accepted) ✓ Read a line for the Player information ✓ and tokenize ✓✓Add a new Player object to the array ✓ at the correct position✓ called constructor with correct parameters ✓ | **10** | |
| 4.4 | **listAllPlayers:** method header correct ✓, appropriate for loop ✓, concatenate the toString ✓ of each object ✓ with a newline ✓ | **5** | |
| 4.5 | **populateMatches:** Method header correct and returns a String ✓ appropriate loop to fill matches array ✓ create new match object at consecutive position ✓opponent object's position determined correctly✓ player objects passed from player array as parameters ✓with correct generated matchcode✓ into match constructor. Each match's toString concatenated and returned ✓ | **7** | |
| 4.6 | **findMatch:** method takes string parameter ✓ Loops through all matches ✓ compare value of parameter✓ with match's code using getMatchCode method ✓ returns a Match object ✓ | **5** | |
| | | | |
| 5.1 | Create an interface/form/class/unit called **TennisUI** ✓ | **1** | |
| 5.2 | Instantiate a **Tournament Object** ✓ (Must have filename parameter, if not penalise -1) | **1** | |
| 5.3 | Methods called✓ and output displayed✓ | **2** | |
| | | | |
| 6.1 | Open a file for reading ✓<br>Find the correct match based on the parameter passed to the findMatch method ✓✓<br>Any variable(s) or data structure for storing each player's games✓ and sets✓ declared and initialised<br>(For example arrays, 2 variables for games, 2 variables for sets, inheritance, extra attributes)<br>Loop while there are lines in the file ✓<br>Working out which player needs a point ✓✓<br>Code for increasing games and sets ✓✓<br>(1 mark for any code that takes the player number of each line in the file and increases games and 1 mark for successfully increasing the number of sets)<br>Resolving sets where the margin must be two games to win✓<br>Code for testing which player wins ✓<br>Generating and setting the score string ✓ | **13** | |
| 6.2 | Method called in interface and output displayed ✓ | **1** | |