



NATIONAL SENIOR CERTIFICATE EXAMINATION
NOVEMBER 2015

INFORMATION TECHNOLOGY: PAPER II

MARKING GUIDELINES

Time: 3 hours

120 marks

These marking guidelines are prepared for use by examiners and sub-examiners, all of whom are required to attend a standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' scripts.

The IEB will not enter into any discussions or correspondence about any marking guidelines. It is acknowledged that there may be different views about some matters of emphasis or detail in the guidelines. It is also recognised that, without the benefit of attendance at a standardisation meeting, there may be different interpretations of the application of the marking guidelines.

SECTION A SQL ANSWER SHEET

QUESTION 1

1.1:

```
SELECT Country, FullName ✓  
FROM tblPlayers ✓  
ORDER BY Country, Fullname ✓
```

1.2:

```
SELECT * ✓  
FROM tblFinals ✓  
WHERE FinalYear < 2000 ✓
```

1.3:

```
DELETE * ✓  
FROM tblPlayers  
WHERE PlayerID = 101 ✓
```

1.4:

```
UPDATE tblTournaments ✓  
SET PrizeMoneyZAR ✓ = PrizeMoney * 11.51 ✓
```

1.5:

```
SELECT MONTH(DateOfBirth) AS BirthMonth ✓, COUNT(*) AS NumBirthdays ✓  
FROM tblPlayers  
GROUP BY ✓ MONTH(DateOfBirth) ✓  
MYSQL: GROUP BY ✓ BirthMonth ✓
```

1.6:

```
SELECT Country ✓, COUNT(*) ✓ AS TotalPlayers  
FROM tblPlayers  
GROUP BY Country ✓  
HAVING ✓ COUNT(*) > 10 ✓  
ORDER BY COUNT(*) DESC ✓  
  
MYSQL:      HAVING ✓ TotalPlayers > 10 ✓  
            ORDER BY TotalPlayers DESC ✓
```

1.7:

```
SELECT TournamentName ✓  
FROM tblTournaments  
WHERE PrizeMoney ✓ > (SELECT AVG(PrizeMoney) ✓ FROM tblTournaments) ✓
```

1.8:

```
SELECT FullName, TournamentName, FinalYear✓  
FROM tblFinals, tblPlayers, tblTournaments✓✓  
WHERE tblFinals.TournamentID = tblTournaments.TournamentID✓  
AND tblFinals.WinnerID = tblPlayers.PlayerID✓  
AND FinalYear MOD 4 = 0✓ AND FinalYear MOD 400 <> 0✓ AND FinalYear > 1995✓
```

OR:

```
SELECT FullName, TournamentName, FinalYear✓  
FROM tblFinals INNER JOIN tblPlayers✓ ON tblFinals.WinnerID = tblPlayers.PlayerID✓  
INNER JOIN tblTournaments✓ ON tblFinals.TournamentID = tblTournaments.TournamentID✓  
WHERE FinalYear MOD 4 = 0✓ AND FinalYear MOD 400 <> 0✓ AND FinalYear > 1995✓
```

(or any solution that works for future years past 2015 and any calculation equivalent to the MOD operator)

1.9:

```
INSERT INTO tblFinals✓ ( TournamentID, FinalYear, FinalName, WinnerID, RunnerUpID )✓  
SELECT✓ 2✓, FinalYear, FinalName, WinnerID, RunnerUpID✓  
FROM tblFinals  
WHERE TournamentID = 1✓ AND FinalYear = 2015✓
```

[40]

SECTION B OBJECT- ORIENTED PROGRAMMING

JAVA SOLUTION:

QUESTION 2

```
// Question 2.1
public class Player✓
{
    // Question 2.2
    private int seed;
    private String fullName;
    private String country;
}
} ✓ Private
} ✓ Correct Types
} ✓ Appropriate names

// Question 2.3
public Player✓(int inSeed, String inFullName, String inCountry)✓
{
    seed = inSeed;
    fullname = inFullName;
    country = inCountry;
}
} ✓ Assign correct values to attributes

// Question 2.4
public String getCountryCode()✓
{
    if (country.indexOf(" ") > -1)✓
    {
        return (" " + country.charAt(0) + country.substring(country.indexOf(" "
        ) + 1, country.indexOf(" ") + 3)).toUpperCase();✓
    }

    return country.substring(0, 3).toUpperCase();✓
}

// Question 2.5
public String toString()✓
{
    return fullname + " (" + getCountryCode()✓ + ")";✓
}
}
```

[14]

QUESTION 3

```
// Question 3.1
public class Match✓
{
    // Question 3.2
    private String matchCode;
    private Player player1;
    private Player player2;
    private String score;
}

// Question 3.3
public Match(String inMatchCode✓, Player inPlayer1, Player inPlayer2✓)
{
    matchCode = inMatchCode;
    player1 = inPlayer1;
    player2 = inPlayer2;
    score = "X";✓
}

// Question 3.4
public String getMatchCode()
{
    return matchCode;✓
}

// Question 3.5
public void setScore(String inScore)✓
{
    score = inScore;
}

// Question 3.6
public String toString()✓
{
    String rString;

    rString = matchCode + " " + player1 + " vs. "✓ + player2✓;

    if (score.equals("X"))✓
    {
        rString = rString + " Not Yet Played"; ✓
    }
    else
    {
        rString = rString + " " + score; ✓
    }

    return rString;
}
}
```

✓ matchCode and score declared correctly
✓✓ player1 and player2 declared as Player

✓ Attribute values assigned correctly

[16]

QUESTIONS 4 AND 6.1

```

import java.io.*;
import java.util.Scanner;

// Question 4.1
public class TournamentManager ✓
{
    // Question 4.2
    private Player[]✓ players = new Player[32];
    private Match[]✓ matches = new Match[16];
    // Question 4.3
    public TournamentManager(String inFileName) ✓
    {
        try
        {
            Scanner scanIn = new Scanner(new FileReader(inFileName)); ✓

            for(int loop = 0; loop < 32; loop++)✓
            {
                String line = scanIn.nextLine();✓

                Scanner scanTokens = new Scanner(line).useDelimiter(",");✓

                int seed = scanTokens.nextInt();✓
                String playerName = scanTokens.next();
                String country = scanTokens.next();

                players[loop]✓ = new Player✓(seed, playerName, country✓);
            }

            scanIn.close();
        }
        catch (FileNotFoundException fne)
        {
            System.out.println("File not found"); ✓
        }
    }

    // Question 4.4
    public String listAllPlayers()✓
    {
        String rString = "";

        for(int loop = 0; loop < players.length; loop++)✓
        {
            rString = rString + players[loop]✓ + "\n"✓;
        }

        return rString✓;
    }

    // Question 4.5
    public String populateMatches()✓
    {
        String rString = "";

        for(int loop = 0; loop < matches.length; loop++)✓
        {
            int opponent = 32 - loop - 1;✓
            matches[loop]✓ = new Match("M" + (loop+1) + (opponent+1)✓,
                players[loop], players[opponent]✓);
            rString = rString + matches[loop] + "\n"✓;
        }
    }
}
    
```

} ✓ Attributes are private
 ✓ Sizes of both arrays are correct

```

    }

    return rString;
}

// Question 4.6
public Match findMatch(String matchCode)✓
{
    for(int loop = 0; loop < matches.length; loop++)✓
    {
        if (matches[loop].getMatchCode().equals(matchCode))✓✓
        {
            return matches[loop];✓
        }
    }

    return null;
}

```

[32]

```

// Question 6.1
public String processResults() throws FileNotFoundException
{
    Scanner resultsIn = new Scanner(new FileReader("results.txt"));✓

    String matchCode = resultsIn.next();
    Match match = findMatch(matchCode);✓ ✓

    int[] games = new int[2];✓
    int[] sets = new int[2]; ✓

    games[0] = games[1] = sets[0] = sets[1] = 0;

    String score = "";

    while (resultsIn.hasNext() && sets[0] != 3 && sets[1] != 3)✓
    {
        int playerIndex = resultsIn.nextInt() - 1;✓✓
        games[playerIndex]++; ✓

        ✓
        if ((games[playerIndex] >= 6) && (games[playerIndex] -
            games[(playerIndex+1) % 2] >= 2))✓
        {
            sets[playerIndex]++;
            score = score + games[0] + "-" + games[1] + " ";
            games[0] = games[1] = 0;
        }

        ✓
        if (sets[playerIndex] == 3)
        {
            score = score + "Player " + playerIndex + " wins";
        }
    }

    resultsIn.close();

    match.setScore(score);✓
    return match.toString();
}
}

```

[13]

QUESTIONS 5 AND 6.2

```
// Question 5.1
public class TennisUI ✓
{
    // Question 5.2
    private static TournamentManager tArr = new
        TournamentManager("players.txt"); ✓

    public static void main(String[] args) throws Exception
    {
        // Question 5.3
        System.out.println("All Players\n=====\n" +
            tArr.listAllPlayers());
        System.out.println("\nAll Matches\n=====\n" +
            tArr.populateMatches());
    }

    // Question 6.2
    System.out.println("\nMatchResult : " + tArr.processResults()); ✓
}
}
```

} ✓ Methods called
✓ Output displayed
[4]

[1]

ALTERNATE JAVA SOLUTION FOR QUESTION 4.3

```
public TournamentManager(String inFileName) ✓
{
    try ✓
    {
        BufferedReader fileIn = new BufferedReader(new
            FileReader(inFileName)); ✓

        for(int loop = 0; loop < 32; loop++) ✓
        {
            String line = fileIn.readLine(); ✓

            String tokens[] = line.split(","); ✓

            int seed = Integer.parseInt(tokens[0]); ✓
            String playerName = tokens[1];
            String country = tokens[2];

            players[loop] ✓ = new Player ✓ (seed, playerName, country ✓);
        }

        fileIn.close();
    }
    catch (FileNotFoundException fne)
    {
        System.out.println("File not found");
    }
    catch (IOException fne)
    {
        System.out.println("File not found");
    }
}
```

[10]

ALTERNATE JAVA SOLUTION FOR QUESTION 6.1

Match.java

```
public class Match
{
    protected String matchCode;
    protected Player player1;
    protected Player player2;
    protected String score;

    ...

    public Player getPlayer1()
    {
        return player1;
    }

    public Player getPlayer2()
    {
        return player2;
    }
}
```

PlayedMatch.java

```
public class PlayedMatch extends Match
{
    private int[] games = new int[2]; ✓
    private int[] sets = new int[2]; ✓

    public PlayedMatch(String inMatchCode, Player inPlayer1, Player inPlayer2)
    {
        super(inMatchCode, inPlayer1, inPlayer2);
        games[0] = games[1] = sets[0] = sets[1] = 0;
    }

    public void playPoint(int player)
    {
        games[player-1]++; ✓
        ✓
        if ((games[player-1] >= 6) && (games[player-1] - games[(player) % 2]
            >= 2) ✓)
        {
            sets[player-1]++;
            score = score + games[0] + "-" + games[1] + " ";
            games[0] = games[1] = 0;
        }
        ✓
        if (sets[player-1] == 3)
        {
            score = score + "Player " + player + " wins";
        }
    }

    public int getSets(int player)
    {
        return sets[player-1];
    }
}
```

✓ Set Score

TournamentManager.java

```
public String processResults() throws FileNotFoundException
{
    Scanner resultsIn = new Scanner(new FileReader("results.txt"));✓
    String matchCode = resultsIn.next();
    Match match = findMatch(matchCode); ✓✓
    PlayedMatch pMatch = new PlayedMatch(match.getMatchCode(),
        match.getPlayer1(), match.getPlayer2());
    pMatch.setScore("");
    while (resultsIn.hasNext() && pMatch.getSets(1) != 3 &&
        pMatch.getSets(2) != 3)✓
    {
        int playerIndex = resultsIn.nextInt();✓✓
        pMatch.playPoint(playerIndex);
    }
    resultsIn.close();
    return pMatch.toString();
}
```

[13]

DELPHI SOLUTION:

QUESTION 2

```

unit uPlayer;

interface

uses SysUtils;

// Question 2.1
type TPlayer = class ✓
    // Question 2.2
    private
        seed : integer;
        fullname : string;
        country : string;
    public
        constructor Create(inSeed : integer; inFullname, inCountry : String);
        function getCountryCode : String;
        function toString : String;
end;

implementation

{ TPlayer }

// Question 2.3
constructor TPlayer.Create (inSeed : integer; inFullname, inCountry :
                            String); ✓
begin
    seed := inSeed;
    fullname := inFullname;
    country := inCountry;
end;

// Question 2.4
function TPlayer.getCountryCode: String; ✓
begin
    if (Pos(' ', country) > 0) then ✓
        begin
            Result := UpCase(country[1]) + UpperCase(Copy(country, Pos(' ', country) +
                1, 2)); ✓
        end
    else
        begin
            Result := UpperCase(Copy(country, 1, 3)); ✓
        end;
    end;
end;

// Question 2.5
function TPlayer.toString: String; ✓
begin
    Result := fullname + ' (' + getCountryCode ✓ + ')'; ✓
end;

end.

```

} ✓ Private
 } ✓ Correct Types
 } ✓ Appropriate names

} ✓ Assign correct values to attributes

[14]

QUESTION 3

```

unit uMatch;

interface

uses uPlayer;

// Question 3.1
type TMatch = class ✓
    // Question 3.2
    private
        matchCode : String;
        player1, player2 : TPlayer;
        score : String;
    public
        constructor Create(inMatchCode: String; inPlayer1, inPlayer2 : TPlayer);
        procedure setScore(inScore : String);
        function getMatchCode : String;
        function toString : String;
end;

implementation

{ TMatch }

// Question 3.3
constructor TMatch.Create (inMatchCode: String; ✓ inPlayer1, inPlayer2:
                                TPlayer✓);
begin
    matchCode := inMatchCode;
    player1 := inPlayer1;
    player2 := inPlayer2;
    score := 'X'; ✓
end;

// Question 3.4
function TMatch.getMatchCode : String; ✓
begin
    Result := matchCode;
end;

// Question 3.5
procedure TMatch.setScore(inScore: String); ✓
begin
    score := inScore;
end;

// Question 3.6
function TMatch.toString: String; ✓
begin
    Result := matchCode + ' ' + player1.toString + ' vs. ' ✓ + player2.toString✓;

    if (score = 'X') then ✓
        begin
            Result := result + ' Not Yet Played'; ✓
        end
    else
        begin
            Result := result + ' ' + score; ✓
        end;
end;

end.
    
```

✓ matchCode and score declared correctly
 ✓✓ player1 and player2 declared as TPlayer

✓ Attribute values assigned correctly

QUESTIONS 4 AND 6.1

```

unit uTournamentManager;

interface

uses uPlayer, uMatch, SysUtils, Classes, Dialogs;

// Question 4.1
type TTournamentManager = class ✓
    private✓
        // Question 4.2
        players : array[1..32] of TPlayer✓;
        matches : array[1..16] of TMatch✓;
    public
        constructor Create(filename : String);
        function listAllPlayers : String;
        function populateMatches : String;
        function findMatch(matchCode : String) : TMatch;
        function processResults : String;
end;

implementation

{ TTournamentManager }

// Question 4.3
constructor TTournamentManager.Create(filename: String); ✓
var
    inFile : textfile;
    curLine : String;
    playerName, country : String;
    seed : integer;
    loop : integer;
begin
    if (FileExists(filename)) then
        begin
            AssignFile(inFile, filename); ✓
            Reset(inFile);

            for loop := 1 to 32 do✓
                begin
                    ReadLn(infile, curLine); ✓

                    ✓✓
                    seed := StrToInt(Copy(curLine, 1, Pos(',', curLine) - 1));
                    Delete(curLine, 1, Pos(',', curLine));

                    playerName := Copy(curLine, 1, Pos(',', curLine) - 1);
                    Delete(curLine, 1, Pos(',', curLine));

                    country := curLine;

                    players[loop]✓ := TPlayer.Create✓(seed, playerName, country✓);
                end;

                CloseFile(infile);
            end
        else
            ShowMessage('File not found'); ✓
        end;
end;

```

✓ Sizes of both arrays are correct

```
// Question 4.4
function TTournamentManager.listAllPlayers: String; ✓
var
  rString : String;
  loop : integer;
begin
  rString := '';

  for loop := 1 to Length(players) do✓
    begin
      rString := rString + players[loop].toString✓ + #13✓;
    end;

  Result := rString; ✓
end;

// Question 4.5
function TTournamentManager.populateMatches : String; ✓
var
  loop, opponent : integer;
  rString : String;
begin
  for loop := 1 to Length(matches) do✓
    begin
      opponent := 32 - loop + 1; ✓
      matches[loop]✓ := TMatch.Create('M' + IntToStr(loop) +
        IntToStr(opponent)✓, players[loop], players[opponent]✓);

      rString := rString + matches[loop].toString + #13;✓
    end;
  Result := rString;
end;

// Question 4.6
function TTournamentManager.findMatch(matchCode: String): TMatch; ✓
var
  loop : integer;
begin
  for loop := 1 to Length(matches) do✓
    begin
      if (matches[loop].getMatchCode = matchCode) then✓✓
        Result := matches[loop]; ✓
    end;
  end;
end;

// Question 6.1
function TTournamentManager.processResults: String;
var
  infile : textfile;
  matchCode, line : String;
  games✓, sets✓ : array[1..2] of Integer;
  match : TMatch;
  score : String;
  playerIndex : Integer;
begin
  AssignFile(infile, 'results.txt'); ✓
  Reset(infile);
  ReadLn(infile, matchCode);

  match := findMatch(matchCode); ✓✓

```

[32]

```

games[1] := 0;
games[2] := 0;
sets[1] := 0;
sets[2] := 0;

score := '';

while (NOT(EOF(infile)) AND (sets[1] <> 3) AND (sets[2] <> 3))✓ do
    begin
        ReadLn(inFile, line);

        playerIndex := StrToInt(line); ✓✓
        Inc(games[playerIndex]);✓

        ✓
        if ((games[playerIndex] >= 6) AND (games[playerIndex] - games[(playerIndex
            MOD 2) + 1] >= 2)✓) then
            begin
                Inc(sets[playerIndex]);
                score := score + IntToStr(games[1]) + '-' + IntToStr(games[2]) + ' ';
                games[1] := 0;
                games[2] := 0;
            end;

        ✓
        if (sets[playerIndex] = 3) then
            begin
                score := score + 'Player ' + IntToStr(playerIndex) + ' wins';
            end;
        end;

match.setScore(score); ✓

CloseFile(infile);

Result := match.toString;

end;
    
```

[13]

QUESTIONS 5 AND 6.2

```

unit uFrmTennisUI;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, ComCtrls, uTournamentManager;

type
    // Question 5.1
    TFrmTennis = class(TForm) ✓
        rchOutput: TRichEdit;
        procedure FormActivate(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    
```

```
frmTennis: TfrmTennis;  
tArr : TTournamentManager;  
  
implementation  
  
{ $R *.dfm }  
  
procedure TfrmTennis.FormActivate(Sender: TObject);  
begin  
    // Question 5.2  
    tArr := TTournamentManager.Create('players.txt'); ✓  
  
    // Question 5.3  
    rchOutput.Lines.Add('All Players' + #13 + '-----' + #13);  
    rchOutput.Lines.Add(tArr.listAllPlayers);  
    rchOutput.Lines.Add(#13 + 'All Matches' + #13 + '-----' + #13);  
    rchOutput.Lines.Add(tArr.populateMatches);  
  
    // Question 6.2  
    rchOutput.Lines.Add(#13 + 'Match Result :' + tArr.processResults); ✓  
  
end;  
  
end.
```

✓ Methods called
✓ Output displayed

[4]

[1]

ALTERNATE DELPHI SOLUTION FOR QUESTION 6.1

uMatch.pas

```
unit uMatch;

protected
  matchCode : String;
  player1, player2 : TPlayer; // 3
  score : String;
  ...
  function getPlayer1 : TPlayer;
  function getPlayer2 : TPlayer;
end;

...

function TMatch.getPlayer1: TPlayer;
begin
  Result := Player1;
end;

function TMatch.getPlayer2: TPlayer;
begin
  Result := Player2;
end;
```

uPlayedMatch.pas

```
unit uPlayedMatch;

interface

uses uMatch, uPlayer, SysUtils;

type TPlayedMatch = class(TMatch)
  private
    games : array[1..2] of Integer; ✓
    sets : array[1..2] of Integer; ✓
  public
    constructor Create(inMatchCode: String; inPlayer1, inPlayer2 : TPlayer);
    override;
    procedure playPoint(p : Integer);
    function getSets(p : Integer) : Integer;
end;

implementation

{ TPlayedMatch }

constructor TPlayedMatch.Create(inMatchCode: String; inPlayer1,
  inPlayer2: TPlayer);
begin
  inherited Create(inMatchCode, inPlayer1, inPlayer2);
  games[1] := 0;
  games[2] := 0;
  sets[1] := 0;
  sets[2] := 0;
end;

function TPlayedMatch.getSets(p: Integer): Integer;
begin
```

```
Result := sets[p];
end;
```

```
procedure TPlayedMatch.playPoint(p: Integer);
begin
```

```
    Inc(games[p]);✓
```

```
✓
```

```
if ((games[p] >= 6) AND (games[p] - games[(p MOD 2) + 1] >= 2)✓) then
begin
    Inc(sets[p]);
    score := score + IntToStr(games[1]) + '-' + IntToStr(games[2]) + ' ';
    games[1] := 0;
    games[2] := 0;
end;
```

```
✓
```

```
if (sets[p] = 3) then
begin
    score := score + 'Player ' + IntToStr(p) + ' wins';
end;
```

} ✓ Set Score

```
end;
```

```
end.
```

uTournamentManager.pas

```
function TTournamentManager.processResults: String;
```

```
var
```

```
    infile : textfile;
    matchCode, line : String;
    match : TMatch;
    pMatch : TPlayedMatch;
    playerIndex : Integer;
```

```
begin
```

```
✓
```

```
AssignFile(infile, 'results.txt');
Reset(inFile);
ReadLn(infile, matchCode);
```

```
match := findMatch(matchCode); ✓✓
```

```
pMatch := TPlayedMatch.Create(matchCode, match.getPlayer1, match.getPlayer2);
```

```
pMatch.setScore('');
```

```
while (NOT(EOF(infile)) AND (pMatch.getSets(1) <> 3) AND (pMatch.getSets(2) <>
3)) ✓ do
```

```
begin
```

```
    ReadLn(inFile, line);
```

```
    playerIndex := StrToInt(line); ✓✓
    pMatch.playPoint(playerIndex);
```

```
end;
```

```
CloseFile(infile);
```

```
Result := pMatch.toString;
```

```
end;
```

[13]

OUTPUT

SECTION A STRUCTURED QUERY LANGUAGE

QUESTION 1.1

Question1_1	
Country	FullName
ARG	David Nalbandian
ARG	Gabriela Sabatini
ARG	Gastón Gaudio
ARG	Guillermo Coria
ARG	Juan Martín del Potro
ARG	Mariano Puerto
AUS	Lleyton Hewitt
AUS	Mark Philippoussis
AUS	Patrick Rafter
AUS	Samantha Stosur
AUT	Thomas Muster
BEL	Justine Henin
BEL	Kim Clijsters
BLR	Victoria Azarenka
BRA	Gustavo Kuerten
CAN	Eugenie Bouchard
CHI	Fernando González
CHI	Marcelo Ríos
CHN	Li Na
CRO	Goran Ivanišević
CRO	Iva Majoli
CRO	Marin Čilić
CYP	Marcos Baghdatis
CZE	Jana Novotná
CZE	Petr Korda
CZE	Petra Kvitová
CZE	Tomáš Berdych
DEN	Caroline Wozniacki
ECU	Andrés Gómez
ESP	Albert Costa
ESP	Alberto Berasategui
ESP	Àlex Corretja
ESP	Arantxa Sánchez Vicario
ESP	Carlos Moyá
ESP	Carlos Moyà
ESP	Conchita Martínez
ESP	David Ferrer
ESP	Juan Carlos Ferrero
ESP	Rafael Nadal

Question1_1	
Country	FullName
ESP	Sergi Bruguera
FRA	Amélie Mauresmo
FRA	Arnaud Clément
FRA	Cédric Pioline
FRA	Jo-Wilfried Tsonga
FRA	Marion Bartoli
FRA	Mary Pierce
FRA	Nathalie Tauziat
FRG	Steffi Graf
GBR	Andy Murray
GBR	Greg Rusedski
GER	Anke Huber
GER	Boris Becker
GER	Michael Stich
GER	Rainer Schüttler
GER	Sabine Lisicki
ITA	Francesca Schiavone
ITA	Sara Errani
JPN	Kei Nishikori
NED	Martin Verkerk
NED	Richard Krajicek
POL	Agnieszka Radwańska
ROU	Simona Halep
RUS	Anastasia Myskina
RUS	Dinara Safina
RUS	Elena Dementieva
RUS	Marat Safin
RUS	Maria Sharapova
RUS	Svetlana Kuznetsova
RUS	Vera Zvonareva
RUS	Yevgeny Kafelnikov
SER	Jelena Janković
SRB	Ana Ivanovic
SRB	Novak Djokovic
SUI	Martina Hingis
SUI	Roger Federer
SUI	Stanislas Wawrinka
SVK	Dominika Cibulková
SWE	Magnus Norman
SWE	Robin Söderling

Question1_1	
Country	FullName
SWE	Stefan Edberg
SWE	Thomas Enqvist
SWE	Thomas Johansson
TCH	Helena Suková
TCH	Ivan Lendl
UKR	Andrei Medvedev
USA	Andre Agassi
USA	Andy Roddick
USA	Jennifer Capriati
USA	Jim Courier
USA	Lindsay Davenport

Question1_1	
Country	FullName
USA	MaliVai Washington
USA	Martina Navratilova
USA	Mary-Joe Fernandez
USA	Michael Chang
USA	Pete Sampras
USA	Serena Williams
USA	Todd Martin
USA	Venus Williams
USA	Zina Garrison
YUG	Monica Seles

QUESTION 1.2

Question1_2				
TournamentID	FinalYear	FinalName	WinnerID	RunnerUpID
1	1990	Women's Singles	16	70
1	1990	Men's Singles	2	1
2	1990	Men's Singles	30	6
2	1990	Women's Singles	17	16
3	1990	Women's Singles	50	16
3	1990	Men's Singles	5	6
4	1990	Women's Singles	15	96
4	1990	Men's Singles	1	3
1	1991	Men's Singles	3	2
1	1991	Women's Singles	17	57
2	1991	Men's Singles	4	6
2	1991	Women's Singles	17	38
3	1991	Men's Singles	1	4
3	1991	Women's Singles	17	15
4	1991	Men's Singles	53	3
4	1991	Women's Singles	16	50
1	1992	Women's Singles	17	70
1	1992	Men's Singles	4	1
2	1992	Men's Singles	4	7
2	1992	Women's Singles	17	16
3	1992	Men's Singles	1	5
3	1992	Women's Singles	17	38
4	1992	Men's Singles	6	55
4	1992	Women's Singles	16	17
1	1993	Men's Singles	4	1
1	1993	Women's Singles	17	16
2	1993	Women's Singles	16	70
2	1993	Men's Singles	31	4
3	1993	Women's Singles	16	69

Question1_2				
TournamentID	FinalYear	FinalName	WinnerID	RunnerUpID
3	1993	Men's Singles	5	86
4	1993	Men's Singles	5	4
4	1993	Women's Singles	16	57
1	1994	Men's Singles	5	60
1	1994	Women's Singles	16	38
2	1994	Women's Singles	38	18
2	1994	Men's Singles	31	74
3	1994	Women's Singles	38	16
3	1994	Men's Singles	6	53
4	1994	Men's Singles	5	55
4	1994	Women's Singles	56	15
1	1995	Men's Singles	6	5
1	1995	Women's Singles	18	38
2	1995	Women's Singles	16	38
2	1995	Men's Singles	32	29
3	1995	Women's Singles	16	17
3	1995	Men's Singles	5	6
4	1995	Men's Singles	5	3
4	1995	Women's Singles	16	38
1	1996	Women's Singles	17	71
1	1996	Men's Singles	3	29
2	1996	Men's Singles	8	53
2	1996	Women's Singles	16	38
3	1996	Women's Singles	16	17
3	1996	Men's Singles	5	29
4	1996	Men's Singles	54	93
4	1996	Women's Singles	16	38
1	1997	Women's Singles	19	18
1	1997	Men's Singles	5	61
2	1997	Women's Singles	39	19
2	1997	Men's Singles	33	31
3	1997	Men's Singles	44	87
3	1997	Women's Singles	19	51
4	1997	Men's Singles	5	86
4	1997	Women's Singles	19	57
1	1998	Men's Singles	7	62
1	1998	Women's Singles	19	56
2	1998	Men's Singles	34	75
2	1998	Women's Singles	38	17
3	1998	Women's Singles	20	19
3	1998	Men's Singles	44	88
4	1998	Men's Singles	5	55
4	1998	Women's Singles	57	97
1	1999	Men's Singles	8	63
1	1999	Women's Singles	19	24

Question1_2				
TournamentID	FinalYear	FinalName	WinnerID	RunnerUpID
2	1999	Women's Singles	16	19
2	1999	Men's Singles	6	76
3	1999	Women's Singles	22	19
3	1999	Men's Singles	6	60
4	1999	Men's Singles	5	6
4	1999	Women's Singles	20	16

QUESTION 1.3

(No output)

QUESTION 1.4

(No output)

QUESTION 1.5

Question1_5	
BirthMonth	NumBirthdays
1	7
2	9
3	8
4	9
5	5
6	13
7	6
8	10
9	12
10	8
11	4
12	9

QUESTION 1.6

Question1_6	
Country	TotalPlayers
USA	14
ESP	11

QUESTION 1.7

Question1_7
TournamentName
Australian Open
Wimbledon

QUESTION 1.8

Question1_8		
FullName	TournamentName	FinalYear
Monica Seles	Australian Open	1996
Boris Becker	Australian Open	1996
Yevgeny Kafelnikov	French Open	1996
Steffi Graf	French Open	1996
Steffi Graf	Wimbledon	1996
Pete Sampras	Wimbledon	1996
Richard Krajicek	US Open	1996
Steffi Graf	US Open	1996
Justine Henin	Australian Open	2004
Roger Federer	Australian Open	2004
Anastasia Myskina	French Open	2004
Gastón Gaudio	French Open	2004
Svetlana Kuznetsova	Wimbledon	2004
Roger Federer	Wimbledon	2004
Roger Federer	US Open	2004
Maria Sharapova	US Open	2004
Maria Sharapova	Australian Open	2008
Novak Djokovic	Australian Open	2008
Ana Ivanovic	French Open	2008
Rafael Nadal	French Open	2008
Serena Williams	Wimbledon	2008
Roger Federer	Wimbledon	2008
Venus Williams	US Open	2008
Rafael Nadal	US Open	2008
Victoria Azarenka	Australian Open	2012
Novak Djokovic	Australian Open	2012
Rafael Nadal	French Open	2012
Maria Sharapova	French Open	2012
Andy Murray	Wimbledon	2012
Serena Williams	Wimbledon	2012
Serena Williams	US Open	2012
Roger Federer	US Open	2012

QUESTION 1.9

(No output)

SECTION B OBJECT-ORIENTED PROGRAMMING

FINAL OUTPUT:

All Players

Colby Noble (HKO)
Craig Hamilton (MYA)
Dante Franklin (SKO)
Chadwick Baird (IND)
Wallace Forbes (CAM)
Tanek Jimenez (ITA)
Jared Flynn (BUR)
Yasir Rosario (TAI)
Levi Klein (BOT)
Kibo Ford (ASA)
Lamar Carney (CAF)
Dieter Reese (BRU)
Brady Serrano (RWA)
Theodore Snyder (TNE)
Flynn Flynn (MOL)
Alec Sellers (UST)
Oscar Price (THA)
Neville Ball (LES)
Nero Barr (GER)
Yardley Craig (UST)
Daquan Morales (SPA)
Stuart Barrett (MIS)
Hector Hinton (DJI)
Dillon Maddox (LIB)
Nash Powers (AND)
Harper Hubbard (HKO)
Stuart Williams (GIB)
Barrett Roberson (PER)
Knox Olson (SOM)
Kasper Greene (SCO)
Emery Kramer (SAF)
Jason Underwood (POR)

All Matches

M132 Colby Noble (HKO) vs. Jason Underwood (POR) Not Yet Played
M231 Craig Hamilton (MYA) vs. Emery Kramer (SAF) Not Yet Played
M330 Dante Franklin (SKO) vs. Kasper Greene (SCO) Not Yet Played
M429 Chadwick Baird (IND) vs. Knox Olson (SOM) Not Yet Played
M528 Wallace Forbes (CAM) vs. Barrett Roberson (PER) Not Yet Played
M627 Tanek Jimenez (ITA) vs. Stuart Williams (GIB) Not Yet Played
M726 Jared Flynn (BUR) vs. Harper Hubbard (HKO) Not Yet Played
M825 Yasir Rosario (TAI) vs. Nash Powers (AND) Not Yet Played
M924 Levi Klein (BOT) vs. Dillon Maddox (LIB) Not Yet Played
M1023 Kibo Ford (ASA) vs. Hector Hinton (DJI) Not Yet Played
M1122 Lamar Carney (CAF) vs. Stuart Barrett (MIS) Not Yet Played
M1221 Dieter Reese (BRU) vs. Daquan Morales (SPA) Not Yet Played
M1320 Brady Serrano (RWA) vs. Yardley Craig (UST) Not Yet Played
M1419 Theodore Snyder (TNE) vs. Nero Barr (GER) Not Yet Played
M1518 Flynn Flynn (MOL) vs. Neville Ball (LES) Not Yet Played
M1617 Alec Sellers (UST) vs. Oscar Price (THA) Not Yet Played

Match Result : M429 Chadwick Baird (IND) vs. Knox Olson (SOM) 6-0 6-8 6-4 6-3 Player 1 wins