



NATIONAL SENIOR CERTIFICATE EXAMINATION
NOVEMBER 2016

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

1.1

```
SELECT * ✓  
FROM tblItems ✓  
WHERE Retailer = "Home Pride" ✓
```

(3)

1.2

```
SELECT Description, Price ✓  
FROM tblItems  
WHERE Price >= 350 ✓ AND ✓ Price <= 500 ✓  
  
OR: WHERE Price BETWEEN ✓ 350 AND ✓ 500 ✓
```

(4)

1.3

```
SELECT BrideName & " & " & ✓ GroomName AS CoupleName ✓, MONTH(WeddingDate) ✓ AS  
MonthOfWedding  
FROM tblCouples  
ORDER BY MONTH(WeddingDate) ✓  
  
(Accept + instead of & in Access, CONCAT in MySQL and || in JavaDB)
```

(4)

1.4

```
UPDATE tblGifts ✓ SET Available = True ✓  
WHERE ✓ RequestedQuantity > BoughtQuantity ✓;
```

(Alternatively: BoughtQuantity < RequestedQuantity or BoughtQuantity – RequestedQuantity = 0)
(Alternatively: UPDATE tblGifts SET Available = RequestedQuantity > BoughtQuantity)

(4)

1.5

```
SELECT Retailer ✓, AVG(Price) ✓ AS AvgItemPrice  
FROM tblItems  
WHERE Description LIKE ✓ '*Platter*' ✓  
GROUP BY Retailer ✓
```

(5)

1.6

```
SELECT SUM(Price) AS ✓ TotalPrice  
FROM tblItems  
WHERE ItemID ✓ NOT IN ✓ (SELECT ItemID  
FROM tblGifts) ✓✓
```

```
SELECT SUM (Price) AS TotalPrice ✓  
FROM tblItems ✓ LEFT JOIN tblGifts ✓  
ON tblItems.itemID = tblGifts ✓  
WHERE tblGifts.itemID IS NULL ✓
```

(If the join is incorrect award no marks. If the ON clause is correct then award 1 mark).

(5)

1.7

```
SELECT BrideName, Description ✓, RequestedQuantity – BoughtQuantity AS QuantityAvailable ✓  
FROM tblItems, tblGifts, tblCouples ✓✓  
WHERE tblItems.ItemID = tblGifts.ItemID ✓ AND tblCouples.CoupleID = tblGifts.CoupleID ✓  
AND Available = True ✓  
ORDER BY BrideName ✓, RequestedQuantity – BoughtQuantity ✓  
(Accept WHERE... RequestedQuantity > BoughtQuantity instead of Available = True or any other  
working condition)
```

(9)

1.8

<pre>INSERT INTO tblGifts (ItemID, CoupleID, RequestedQuantity, BoughtQuantity, Available) ✓ SELECT ItemID, 9, RequestedQuantity, 0, True } ✓✓ (Hard-coded values) ✓ (Field names in select) FROM tblGifts } ✓(Only award for correct nested SELECT) WHERE CoupleID = 2 ✓</pre>	(6)
--	-----

[40]

SECTION B OBJECT-ORIENTED PROGRAMMING

JAVA SOLUTION

QUESTION 2

```
// Question 2.1
public class Wedding ✓
{
    // Question 2.2
    private String bride;
    private String groom;
    private String weddingDate;
    private String venue;
    private int guests;
}

// Question 2.3
public Wedding ✓ (String inBride, String inGroom, String inDate, String
                 inVenue, int inGuests) ✓
{
    bride = inBride;
    groom = inGroom;
    weddingDate = inDate;
    venue = inVenue;
    guests = inGuests;
}

// Question 2.4
// 3
public String getWeddingDate()
{
    return weddingDate;
}

public String getVenue()
{
    return venue;
}

public int getGuests()
{
    return guests;
}

// Question 2.5
public boolean equals(String inDay, String inVenue) ✓
{
    if (inDay.equals(getWeddingDate()) ✓ && inVenue.equals(getVenue())) ✓
    {
        return true;
    }
    else
    {
        return false;
    }
}
}
```

} ✓ Private
 } ✓ Correct Types
 } ✓ Appropriate names

} ✓ Assign correct values to attributes

} ✓ Accessors named correctly
 } ✓ Return correct type
 } ✓ Return correct attribute value

} ✓ True or false returned correctly

Question 2.5:

- Accept Integer and String types as parameters for the inVenue parameter
- Accept == and .equals when comparing inVenue with the venue property
- Accept any attempt at conversion between integers and strings with regards to inVenue parameter

```
// Question 2.6
public String toString() ✓
{
    return bride + " & " + groom + "\t\t" ✓ + guests + " guests"; ✓
}
}
```

[17]

QUESTION 3

```
// Question 3.1
public class BookingDay ✓
{
    // Question 3.2
    private String bookingDate;
    private Wedding garden;
    private Wedding lake;
}

// Question 3.3
public BookingDay (String inBDate ✓, Wedding inGarden, Wedding inLake✓)
{
    bookingDate = inBDate;
    garden = inGarden;
    lake = inLake;
}

// Question 3.4
public String getOverBooked() ✓
{
    String rString = "";

    if (garden != null && garden.getGuests() > 130)
    {
        rString = "Garden venue is overbooked by " + (garden.getGuests() -
            130) + " guests\n";
    }
    if (lake != null && lake.getGuests() > 160)
    {
        rString = rString + "Lake venue is overbooked by " +
            (lake.getGuests() - 160) + " guests\n"; ✓
    }

    return rString;
}

// Question 3.5
public String toString() ✓
{
    String rString = bookingDate;

    if (garden != null) ✓
    {
        rString = rString + "\nGarden : " + garden.toString();
    }
    if (lake != null)
    {
        rString = rString + "\nLake : " + lake.toString(); ✓
    }

    if (garden == null && lake == null) ✓
    {
        rString = rString + "\nNo weddings booked";
    }

    return rString + "\n" + getOverBooked(); ✓
}
}
```

✓ bookingDate declared correctly
✓✓ garden and lake declared as Wedding
(Penalise -1 again for non-private attributes)

✓ bookingDate assigned correctly
✓ garden and lake assigned correctly
(Penalise for incorrect garden and lake types again)

✓ perform checks in two separate if statements (do not accept if...else) ✓ that garden or lake properties are not null ✓ check the number of guests does not exceed 160 and 130 respectively

[18]

QUESTIONS 4 AND 6.1

```

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

// Question 4.1
public class VenueManager ✓
{
    // Question 4.2
    private Wedding[] ✓ weddings = new Wedding[50];
    private BookingDay[] ✓ bookings = new BookingDay[30]; } ✓ correct size
                                                    ✓ both declared as private

    // Question 4.3
    public VenueManager(String inFilename) ✓
    {
        try ✓
        {
            Scanner inFile = new Scanner(new FileReader(inFilename)); ✓

            for (int loop = 0; loop < 50; loop++) ✓
            {
                String line = inFile.nextLine(); ✓
                Scanner scanTokens = new Scanner(line).useDelimiter("#"); ✓

                weddings[loop] ✓ = new Wedding ✓ (scanTokens.next(),
                    scanTokens.next(), scanTokens.next(), scanTokens.next(),
                    scanTokens.nextInt() ✓); ✓
            }

            inFile.close();

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

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

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

        return rString ✓;
    }

    // Question 4.5
    public Wedding getWeddingsOnDay(String inDay, String inVenue) ✓
    {
        Wedding rWedding = null; ✓

        for (int loop = 0; loop < weddings.length; loop++) ✓
        {
            if (weddings[loop].equals(inDay,
                inVenue)) ✓
            {
                rWedding = weddings[loop]; ✓
            }
        }
    }
}

```

Accept any comparison of the parameters against objects in the array
--

```
        return rWedding;
    }

// Question 4.6
public String processBookings()
{
    String rString = "";

    for (int loop = 0; loop < bookings.length; loop++) ✓
    {
        String bDate = "";
        int day = loop + 1;
        if (day < 10) ✓
        {
            bDate = "2016-11-0" + day; ✓
        }
        else
        {
            bDate = "2016-11-" + day;
        }

        Wedding garden = getWeddingsOnDay(bDate, "Garden"); ✓
        Wedding lake = getWeddingsOnDay(bDate, "Lake"); ✓

        bookings[loop] ✓ = new BookingDay(bDate, garden, lake); ✓
        rString = rString + bookings[loop] + "\n"; ✓
    }

    return rString;
}
```

[33]

```
// Question 6.1
public String displayCalendar(int startDay) ✓
{
    String calendarString = " M Tu W Th F Sa Su\n"; ✓

    for(int i = 1; i < startDay; i++) ✓
    {
        calendarString += " ";
    }

    for (int i = 1; i <= 30; i++) ✓
    {
        if (i < 10) ✓
        {
            calendarString += " " + i;
        }
        else
        {
            calendarString += " " + i;
        }

        if ((i + startDay - 1) % 7 == 0) ✓
        {
            calendarString += "\n"; ✓
        }
    }

    return calendarString;
}
}
```

[7]

QUESTIONS 5 AND 6.2

```
// Question 5.1
public class WeddingUI ✓
{
    public static void main(String[] args)
    {
        // Question 5.2
        VenueManager vArr = new VenueManager("weddings.txt"); ✓

        // Question 5.3
        System.out.println("Weddings\n=====\n" + vArr.listAllWeddings()); ✓
        System.out.println("Bookings\n=====\n" + vArr.processBookings()); ✓ [4]

        // Question 6.2
        System.out.println("Calendar\n=====\n" + vArr.displayCalendar(7)); ✓ [1]
    }
}

[7]
```

DELPHI SOLUTION:

SECTION B OBJECT-ORIENTED PROGRAMMING

QUESTION 2

```
unit uWedding;

interface

uses SysUtils;

// Question 2.1
type TWedding = class✓
  // Question 2.2
  private
    bride : String;
    groom : String;
    weddingDate : String;
    venue : String;
    guests : integer;
  public
    constructor Create(inBride, inGroom, inDate, inVenue : String; inGuests :
      integer);
    function getWeddingDate : String;
    function getVenue : String;
    function getGuests : integer;
    function equals(inDay, inVenue : String) : boolean;
    function toString : String;
end;

implementation

{ TWedding }

// Question 2.3
constructor TWedding.Create✓(inBride, inGroom, inDate, inVenue: String;
  inGuests: integer); ✓
begin
  bride := inBride;
  groom := inGroom;
  weddingDate := inDate;
  venue := inVenue;
  guests := inGuests;
end;

// Question 2.4
function TWedding.getGuests: integer;
begin
  Result := guests;
end;

function TWedding.getVenue: String;
begin
  Result := venue;
end;

function TWedding.getWeddingDate: String;
begin
  Result := weddingDate;
end;
```

✓Private
✓Correct Types
✓Appropriate names

✓ Assign correct values to attributes

✓ Accessors named correctly
✓ Return correct type
✓ Return correct attribute value

```
// Question 2.5
function TWedding.equals(inDay, inVenue: String): boolean; ✓
begin
  if (inDay = getWeddingDate) ✓ AND (inVenue = getVenue) ✓ then
    begin
      Result := true;
    end
  else
    begin
      Result := false;
    end;
  end;
end;
```

✓ true or false returned correctly

Question 2.5:

- Accept Integer and String types as parameters for the inVenue parameter
- Accept == and .equals when comparing inVenue with the venue property
- Accept any attempt at conversion between integers and strings with regards to inVenue parameter

```
// Question 2.6
function TWedding.toString: String; ✓
begin
  Result := bride + ' & ' + groom + #9 + #9 ✓ + IntToStr(guests) + ' guests'; ✓
end;

end.
```

[17]

QUESTION 3

```
unit uBookingDay;

interface

uses SysUtils, uWedding;

// Question 3.1
type TBookingDay = class✓
  // Question 3.2
  private
    bookingDate : String;
    garden : TWedding;
    lake : TWedding;
  public
    constructor Create(inBDate : String; inGarden, inLake : TWedding);
    function getOverBooked : String;
    function toString : String;
end;

implementation

{ TBookingDay }

// Question 3.3
constructor TBookingDay.Create (inBDate: String;✓ inGarden, inLake: TWedding✓);
begin
  bookingDate := inBDate;
  garden := inGarden;
  lake := inLake;
end;

// Question 3.4
function TBookingDay.getOverBooked: String; ✓
begin
  Result := '';

  if (garden <> nil) AND (garden.getGuests > 130) then
    begin
      Result := 'Garden venue is overbooked by ' + IntToStr(garden.getGuests -
        130) + ' guests' + #13;
    end;
  if (lake <> nil) AND (lake.getGuests > 160) then
    begin
      Result := 'Lake venue is overbooked by ' + IntToStr(lake.getGuests - 160)
        + ' guests' + #13; ✓
    end;
end;
end;
```

✓ bookingDate declared correctly
✓✓ garden and lake declared as Wedding
(Penalise -1 again for non-private attributes)

✓ bookingDate assigned correctly
✓ garden and lake assigned correctly
(Penalise for incorrect garden and lake types again)

✓ perform checks in two separate if statements (do not accept if...else) ✓ that garden or lake properties are not null ✓ check the number of guests does not exceed 160 and 130 respectively

```
// Question 3.5
function TBookingDay.toString: String; ✓
begin
  Result := bookingDate;

  if (garden <> nil) then ✓
  begin
    Result := Result + #13 + 'Garden : ' + garden.toString;
  end;
  if (lake <> nil) then
  begin
    Result := Result + #13 + 'Lake   : ' + lake.toString; ✓
  end;

  if (garden = nil) AND (lake = nil) then ✓
  begin
    Result := Result + #13 + 'No weddings booked';
  end;

  Result := Result + #13 + getOverBooked(); ✓
end;

end.
```

[18]

QUESTIONS 4 AND 6.1

```
unit uVenueManager;

interface

uses SysUtils, Classes, Dialogs, uWedding, uBookingDay;

// Question 4.1
type TVenueManager = class ✓
  // Question 4.2
  private
    weddings : array[1..50] of TWedding; ✓
    bookings : array[1..30] of TBookingDay; ✓ } ✓ correct size
                                                ✓ both declared as private
  public
    constructor Create(inFilename : String);
    function listAllWeddings : String;
    function getWeddingsOnDay(inDay, inVenue : String) : TWedding;
    function processBookings : String;
    function displayCalendar(startDay : integer) : String;
end;

implementation

{ TVenueManager }
```

```
// Question 4.3
constructor TVenueManager.Create(inFilename: String); ✓
var
    inFile : textfile;
    curLine : String;
    bride, groom, weddingDate, venue : String;
    guests : integer;
    loop : integer;
begin
    if (FileExists(inFilename)) then ✓
        begin
            AssignFile(inFile, inFilename); ✓
            Reset(inFile);

            for loop := 1 to 50 do ✓
                begin
                    ReadLn(infile, curLine); ✓
                    ✓✓
                    bride := Copy(curLine, 1, Pos('#', curLine) - 1);
                    Delete(curLine, 1, Pos('#', curLine));

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

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

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

                    guests := StrToInt(curLine);

                    weddings[loop] ✓ := TWedding.Create ✓ (bride, groom, weddingDate,
                                                                venue, guests); ✓
                end;

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

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

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

    Result := rString; ✓
end;
```

```
// Question 4.5
function TVenueManager.getWeddingsOnDay(inDay, inVenue: String): TWedding; ✓
var
    rWedding : TWedding;
    loop : integer;
begin
    rWedding := nil; ✓

    for loop := 1 to length(weddings) do ✓
        begin
            if (weddings[loop].equals(inDay,
                inVenue)) then ✓
                begin
                    rWedding := weddings[loop]; ✓
                end;
            end;

        Result := rWedding;
    end;
```

Accept any comparison of the parameters against objects in the array

```
// Question 4.6
function TVenueManager.processBookings: String;
var
    rString, bDate : String;
    garden, lake : TWedding;
    loop : integer;
begin
    rString := '';

    for loop := 1 to length(bookings) do ✓
        begin
            bDate := '';

            if (loop < 10) then ✓
                begin
                    bDate := '2016-11-0' + IntToStr(loop); ✓
                end
            else
                begin
                    bDate := '2016-11-' + IntToStr(loop);
                end;

            garden := getWeddingsOnDay(bDate, 'Garden'); ✓
            lake := getWeddingsOnDay(bDate, 'Lake'); ✓

            bookings[loop] ✓ := TBookingDay.Create(bDate, garden, lake); ✓
            rString := rString + bookings[loop].toString + #13; ✓
        end;

        Result := rString;
    end;
```

[33]

```
// Question 6.1
function TVenueManager.displayCalendar(startDay : integer): String; ✓
var
    rString : String;
    loop : integer;
begin
    rString := ' M Tu W Th F Sa Su' + #13; ✓

    for loop := 1 to startDay - 1 do ✓
        begin
            rString := rString + '  ';
        end;

    for loop := 1 to 30 do ✓
        begin
            if (loop < 10) then ✓
                begin
                    rString := rString + ' ' + IntToStr(loop);
                end
            else
                begin
                    rString := rString + ' ' + IntToStr(loop);
                end;
            if ((loop + startDay - 1) MOD 7 = 0) then ✓
                begin
                    rString := rString + #13; ✓
                end;
            end;

        Result := rString;
    end;
end.
```

[7]

QUESTIONS 5 AND 6.2

```
unit uFrmWeddingUI;

interface

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

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

var
    frmWeddingUI: TfrmWeddingUI;
    vArr : TVenueManager;

implementation

{$R *.dfm}
```



```

procedure TfrmWeddingUI.FormActivate(Sender: TObject);
begin

    // Question 5.2
    vArr := TVenueManager.Create('weddings.txt'); ✓

    // Question 5.3
    rchOutput.Lines.Add('Weddings' + #13 + '-----' + #13 + vArr.listAllWeddings
        + #13); ✓
    rchOutput.Lines.Add('Bookings' + #13 + '-----' + #13 + vArr.processBookings
        + #13); ✓

    // Question 6.2
    rchOutput.Lines.Add('Calendar' + #13 + '-----' + #13 +
        vArr.displayCalendar(7) + #13); ✓

end;

end.
    
```

[4]

[1]

ALTERNATE SOLUTION FOR QUESTION 6.1

```

function TVenueManager.displayCalendar(startDay : integer): String; ✓
var
    rString : String;
    loop : integer;
    myDateTime : TDateTime;
begin
    rString := ' M Tu W Th F Sa Su' + #13; ✓

    for loop := 1 to startDay - 1 do ✓
        begin
            rString := rString + '  ';
        end;

    myDateTime := StrToDateTime('2016-11-01');

    for loop := 1 to 30 do ✓
        begin
            if (DayOfTheMonth(myDateTime) < 10) then ✓
                begin
                    rString := rString + ' ' + IntToStr(DayOfTheMonth(myDateTime));
                end
            else
                begin
                    rString := rString + ' ' + IntToStr(DayOfTheMonth(myDateTime));
                end;

            if ((loop + startDay - 1) MOD 7 = 0) then ✓
                begin
                    rString := rString + #13; ✓
                end;

            myDateTime := IncDay(myDateTime, 1);
        end;

    Result := rString;
end;
    
```

[7]

OUTPUT

SECTION A STRUCTURED QUERY LANGUAGE

QUESTION 1.1

Query1			
ItemID	Description	Price	Retailer
12446	Tray Initial Set	R274.95	Home Pride
13738	Server Set:Classic	R164.95	Home Pride
18095	Salad Server:Long	R65.95	Home Pride
18482	Dip Warmer	R164.95	Home Pride
19579	Seafood Aluminium:Crab & Fish	R516.95	Home Pride
20336	Seafood Buffet	R1,099.95	Home Pride
22512	Brushed Bronze Lantern	R329.95	Home Pride
23156	Platter:FDL Aluminium	R1,759.95	Home Pride
23723	Candle:FDL	R769.95	Home Pride
24962	Large Name Plaque	R549.95	Home Pride
25685	Tile:Wedding Horizontal	R439.95	Home Pride
26151	Chip & Dip:Fishy Fishy	R494.95	Home Pride
27580	Monkey Bread Cooker	R329.95	Home Pride
28060	Organic Pearl Square Dish	R549.95	Home Pride
28850	Chowder Bowl:Sea Catch	R120.95	Home Pride
28853	Ice Cream Bowl:Sea Catch	R120.95	Home Pride
30347	Frame:Horiz Scroll	R362.95	Home Pride
30710	Pesto Gift Set	R351.95	Home Pride
30889	Table Runner	R274.95	Home Pride
30948	Mr Mrs Hand Towel Set	R219.95	Home Pride
30967	Koozie Set:Bride & Groom	R219.95	Home Pride
31036	Pitcher	R329.95	Home Pride
31260	Cutting Board	R164.95	Home Pride
31556	Press Tile Frame	R296.95	Home Pride
31557	Votive Holder	R131.95	Home Pride
31559	Vase Set	R296.95	Home Pride
31628	Album:Pearl Tassle	R1,429.95	Home Pride
31735	Vase:Green Glaze	R131.95	Home Pride

QUESTION 1.2

Query2	
Description	Price
Beaded Aluminium:Platter Sqaure	R472.95
Beaded Aluminium:Bowl 28 cm Square	R384.95
Bucket:Round With handles	R494.95
Seafood Aluminium:Crab Bowl	R384.95
Vase:Verde Round	R406.95
Highball Glass:initial	R406.95
Tile:Wedding Horizontal	R439.95
Chip & Dip:Fishy Fishy	R494.95
Pillow:Fleur De igre	R461.95
Frame:Horiz Scroll	R362.95
Pesto Gift Set	R351.95
Square Crab Glass	R439.95
Eyelet Bowl Set	R384.95
Frame:Our Wedding	R384.95
Frame:Our Honeymoon	R351.95
Pitcher:Large Blue Trim	R439.95

QUESTION 1.3

Query3	
CoupleName	MonthOfWedding
Wanda Ferguson & Albert Richardson	4
Susan Smith & Ryan Jones	7
Kathy Hawkins & Joshua Stevens	8
Lori Brown & Carlos Diaz	10
Linda Baker & Eric Cook	11
Sarah Carroll & Stephen Lawson	12
Marie Adams & Timothy Adams	12
Diane Kennedy & Andrew Woods	12
Emily Ryan & Harold Grant	12

QUESTION 1.4

(No output)

QUESTION 1.5

Query5	
Retailer	AvgItemPrice
Decor Centre	R511.45
Furnmart	R274.95
Home Pride	R1,759.95

QUESTION 1.6

Query6
TotalPrice
R5,609.15

QUESTION 1.7

Query7		
BrideName	Description	QuantityAvailable
Diane Kennedy	Monkey Bread Cooker	1
Diane Kennedy	Seafood Aluminium:Crab Bowl	1
Diane Kennedy	Pitcher:Acrylic	1
Diane Kennedy	Square Crab Glass	1
Diane Kennedy	Vase Set	1
Diane Kennedy	Platter:FDL Aluminium	2
Diane Kennedy	Crawfish Tumbler	2
Diane Kennedy	Vase:Verde Round	2
Diane Kennedy	Tile:Wedding Damask	3
Emily Ryan	Fleur De Lis Pillow	1
Emily Ryan	Crab Tumbler	1
Emily Ryan	Ice Cream Bowl:Sea Catch	1
Emily Ryan	Platter:Sea Catch	1
Emily Ryan	Party Tub:Acrylic	1
Emily Ryan	Tile:Wedding Horizontal	1
Emily Ryan	Chip & Dip:34 cm Beaded	2
Emily Ryan	Finial:FDL Lg Iron	2
Emily Ryan	Tile:Wedding Damask	2
Emily Ryan	Plate:Sea Catch	3
Emily Ryan	Votive Holder	4
Kathy Hawkins	Bucket:Round With handles	1
Kathy Hawkins	Server Set:Classic	1
Kathy Hawkins	Table Runner	1
Kathy Hawkins	Koozie Set:Bride & Groom	1
Kathy Hawkins	Party Tub:Acrylic	1
Kathy Hawkins	Platter:Sea Catch	1
Kathy Hawkins	Beaded Aluminium:Bowl Oval	1
Kathy Hawkins	Pitcher:Acrylic	1
Kathy Hawkins	Plate:Sea Catch	2
Kathy Hawkins	Pitcher:Large Blue Trim	2
Kathy Hawkins	Bowl & Spreader Set	2
Kathy Hawkins	Tibit Set	2
Kathy Hawkins	Frame:Our Wedding	3
Kathy Hawkins	Candle:FDL	3
Kathy Hawkins	Margarita Acrylic	4
Kathy Hawkins	Album:Pearl Tassle	4
Kathy Hawkins	Album:Victoria Brooch	4
Linda Baker	Seafood Aluminium:Crab & Lobster	1
Linda Baker	Seafood Buffet	1
Linda Baker	Tile:Wedding Damask	1
Linda Baker	Platter:Sea Catch	1
Linda Baker	Plate:Sea Catch	1
Linda Baker	Note Pad Set:Initials	1
Linda Baker	Table Runner	2
Linda Baker	Seafood Aluminium:Crab & Fish	2

Query7		
BrideName	Description	QuantityAvailable
Linda Baker	Vase:Verde Round	4
Lori Brown	Press Tile Frame	1
Lori Brown	Ice Cream Bowl:Sea Catch	1
Lori Brown	Pillow:Fleur De igre	2
Lori Brown	Bowl:Reflections	2
Lori Brown	Beaded Alum:Platter Square	2
Lori Brown	Album:Pearl Tassle	3
Lori Brown	Table Runner	4
Lori Brown	Fleur De Lis Pillow	4
Marie Adams	Bowl:Square 3 in 1	1
Marie Adams	Dip Warmer	1
Marie Adams	Wine Holder:Shuffles	1
Marie Adams	Fleur De Lis Pillow	2
Marie Adams	Highball Glass:initial	3
Sarah Carroll	Salad Server:Long	1
Sarah Carroll	Tibit Set	1
Sarah Carroll	Pillow:Fleur De igre	1
Sarah Carroll	Highball Glass:initial	1
Sarah Carroll	Brushed Bronze Lantern	1
Sarah Carroll	Bread Board Napkin Set	2
Sarah Carroll	Square Crab Glass	2
Sarah Carroll	Bowl:Square 3 in 1	2
Sarah Carroll	Ladle:Acrylic	3
Sarah Carroll	Pitcher	3
Sarah Carroll	Wine Holder:Shuffles	3
Wanda Ferguson	Chip & Dip:34 cm Beaded	1
Wanda Ferguson	Brushed Bronze Lantern	1
Wanda Ferguson	Bowl & Spreader Set	1
Wanda Ferguson	Pillow:Fleur De igre	2
Wanda Ferguson	Vase Set	4

QUESTION 1.8

(No output)

SECTION B OBJECT-ORIENTED PROGRAMMING

FINAL OUTPUT:

Weddings

=====

Chiquita Sanford & Magee Sosa	84 guests
Sade Knox & Logan Duncan	101 guests
Kirestin Tyler & Kirk Palmer	146 guests
Ayanna Rivas & Abel Hopkins	139 guests
Hadassah Mason & Carlos Hoffman	120 guests
Jessamine Max & Malcolm Cass	125 guests
Kevin Mcintosh & Kermit Mick	95 guests
Celeste Chandler & Dieter Chaps	99 guests
Lacota Buchanan & Devin Bass	141 guests
Nevada Summers & Brenden Sot	147 guests
Vivian Dillon & Abdul Mcneil	102 guests
Daryl Mccarty & Fred Hubbard	109 guests
Victoria Adams & Ulric Kennedy	139 guests
Nicole Knowles & Stone Hogan	105 guests
Yolanda May & Porter Navarro	100 guests
Lunea Callahan & Aristotle Moon	126 guests
Melodie Burns & Fred Pitout	120 guests
Jaquelyn Hood & Orson Case	138 guests
Alyssa Larson & Trevor Burgess	81 guests
Aye Norman & Brian Daniel	84 guests
Joan Taylor & Keith Marquez	116 guests
Karly Mcfadden & Gray Burgess	80 guests
Mackenzie Chaney & Kasimir Hull	94 guests
Paula Holman & Jo Rogers	135 guests
Melinda Levine & Wayne Strong	96 guests
Gerry Sanders & Erasmus Oneill	95 guests
Lael Wheeler & Andrew Cross	134 guests
Rowan Hawkins & Thor Scott	142 guests
Lynn Finch & Arsenio Pratt	106 guests
Wendy Charles & Brady Rutledge	122 guests
Sydney Travis & Warren Sloan	185 guests
Imani Mueller & Evan Ayala	110 guests
Callie Hendrix & Lamar Lopez	113 guests
Chloe Harris & Slade Osborne	108 guests
Whoopi Blake & Hamish Reyes	147 guests
Rachel Chang & Chaim Coxon	97 guests
Kylie Bird & Carter Berg	100 guests
Zephr Lamb & Hashim wall	145 guests
Quin Mcpherson & Ulysses Grant	165 guests
Courtney Stein & Guy Bates	103 guests
Chanda Olsen & Byron Everett	124 guests
Rhonda Mack & Arsenio Long	100 guests
Vera Hatfield & Yuli Schwartz	144 guests
Mel Mcneil & Keegan Johnston	89 guests
Ingrid Haney & Aristotle Bryant	102 guests
Ocean Rowland & Benedict Vill	138 guests
Bianca Gray & Warren Pollard	100 guests
Ori Molina & Simon Molina	136 guests
Kerry Aguirre & Blaze Petersen	83 guests
Tanya Briggs & Sawyer Hunter	98 guests

Bookings

=====

2016-11-01	
Garden : Lynn Finch & Arsenio Pratt	106 guests
2016-11-02	
Garden : Bianca Gray & Warren Pollard	100 guests
Lake : Mel Mcneil & Keegan Johnston	89 guests
2016-11-03	
Garden : Ingrid Haney & Aristotle Bryant	102 guests
Lake : Kirestin Tyler & Kirk Palmer	146 guests
2016-11-04	
Garden : Callie Hendrix & Lamar Lopez	113 guests
2016-11-05	
Garden : Chiquita Sanford & Magee Sosa	84 guests
Lake : Joan Taylor & Keith Marquez	116 guests
2016-11-06	
Garden : Jessamine Max & Malcolm Cass	125 guests
Lake : Karly Mcfadden & Gray Burgess	80 guests
2016-11-07	
Garden : Daryl Mccarty & Fred Hubbard	109 guests
2016-11-08	
Garden : Vivian Dillon & Abdul Mcneil	102 guests
Lake : Lunea Callahan & Aristotle Moon	126 guests

2016-11-09		
Lake	: Jaquelyn Hood & Orson Case	138 guests
2016-11-10		
Garden	: Celeste Chandler & Dieter Chaps	99 guests
Lake	: Tanya Briggs & Sawyer Hunter	98 guests
2016-11-11		
Lake	: Kerry Aguirre & Blaze Petersen	83 guests
2016-11-12		
Garden	: Kevyn Mcintosh & Kermit Mick	95 guests
Lake	: Nevada Summers & Brenden Sot	147 guests
2016-11-13		
Garden	: Victoria Adams & Ulric Kennedy	139 guests
Lake	: Imani Mueller & Evan Ayala	110 guests
Garden venue is overbooked by 9 guests		
2016-11-14		
Garden	: Vera Hatfield & Yuli Schwartz	144 guests
Lake	: Chanda Olsen & Byron Everett	124 guests
Garden venue is overbooked by 14 guests		
2016-11-15		
No weddings booked		
2016-11-16		
Garden	: Yolanda May & Porter Navarro	100 guests
Lake	: Ocean Rowland & Benedict Vill	138 guests
2016-11-17		
Lake	: Lael Wheeler & Andrew Cross	134 guests
2016-11-18		
Garden	: Rachel Chang & Chaim Coxon	97 guests
Lake	: Paula Holman & Jo Rogers	135 guests
2016-11-19		
Garden	: Rowan Hawkins & Thor Scott	142 guests
Lake	: Wendy Charles & Brady Rutledge	122 guests
Garden venue is overbooked by 12 guests		
2016-11-20		
Garden	: Gerry Sanders & Erasmus Oneill	95 guests
Lake	: Quin Mcpherson & Ulysses Grant	165 guests
Lake venue is overbooked by 5 guests		
2016-11-21		
Garden	: Melodie Burns & Fred Pitout	120 guests
2016-11-22		
Garden	: MacKenzie Chaney & Kasimir Hull	94 guests
Lake	: Ori Molina & Simon Molina	136 guests
2016-11-23		
Lake	: Rhonda Mack & Arsenio Long	100 guests
2016-11-24		
Garden	: Avye Norman & Brian Daniel	84 guests
2016-11-25		
Garden	: Alyssa Larson & Trevor Burgess	81 guests
Lake	: Whoopi Blake & Hamish Reyes	147 guests
2016-11-26		
Garden	: Courtney Stein & Guy Bates	103 guests
2016-11-27		
Garden	: Kylie Bird & Carter Berg	100 guests
Lake	: Sydney Travis & Warren Sloan	185 guests
Lake venue is overbooked by 25 guests		
2016-11-28		
Garden	: Hadassah Mason & Carlos Hoffman	120 guests
Lake	: Chloe Harris & Slade Osborne	108 guests
2016-11-29		
Garden	: Lacota Buchanan & Devin Bass	141 guests
Lake	: Melinda Levine & Wayne Strong	96 guests
Garden venue is overbooked by 11 guests		
2016-11-30		
Garden	: Ayanna Rivas & Abel Hopkins	139 guests
Lake	: Zephr Lamb & Hashim Wall	145 guests
Garden venue is overbooked by 9 guests		

Calendar

=====

M	Tu	W	Th	F	Sa	Su
	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					