NATIONAL SENIOR CERTIFICATE EXAMINATION
MAY 2021

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 STRUCTURED QUERY LANGUAGE

1.1

```
SELECT Paper, StartTime, EndTime
FROM tblTimeTable
WHERE EndTime=#16:30#              ='16:30' for JavaDB and MySQL
```

1.2

```
SELECT *
FROM tblTimeTable
WHERE Paper LIKE "*PII*"       LIKE '%PII%' for JavaDB and MySQL
ORDER BY ExamDate
```

1.3

```
SELECT Firstname, Surname, DateOfBirth
FROM tblStudents
WHERE Gender = "M" AND MONTH(DateOfBirth) = 3
```

1.4

```
SELECT Centre, Gender,
COUNT(*) AS StudentTotal
FROM tblStudents  SELECT and FROM correct
GROUP BY Centre, Gender
```

1.5

```
INSERT INTO tblStudents ( Firstname, Surname, DateOfBirth, Gender, Centre )
  Correct insert and table
  correct attributes
VALUES ("Sarah", "Henderson", #1999-09-23#, "F", 1021)
 VALUES (..) with matching order
 Correct values for all attributes
MySQL and JavaDB – Use single quotes instad of #
```

1.6

```
SELECT PaperID, COUNT(*) AS StudentCount
FROM tblCandidates
WHERE PaperID LIKE "MAT*"   % instead of * for JavaDB and MySQL
GROUP BY PaperID
HAVING COUNT(*) > 7
```

1.7

```
SELECT PaperID
FROM tblTimeTable
WHERE PaperID NOT IN (SELECT PaperID FROM tblCandidates)
```

1.8

```
SELECT Firstname & " " & Surname AS FullName,    MySql : use concat, JavaDB : use ||
MAX(ExamDate) AS LastExam
FROM tblStudents, tblCandidates, tblTimeTable
WHERE tblStudents.StudentID = tblCandidates.StudentID
AND tblCandidates.PaperID = tblTimeTable.PaperID
GROUP BY Firstname & " " & Surname
```

SECTION B OBJECT-ORIENTED PROGRAMMING

JAVA SOLUTION:

QUESTION 2

```
// Question 2.1
public class Exam  {
    // Question 2.2
    private String examDate;
    private String paper;
    private String startTime;
    private double duration;

    // Question 2.3
    public Exam ( String inExamDate, String inPaper,
                        String inStartTime, double inDuration)    {
        examDate = inExamDate;
        paper = inPaper;
        startTime = inStartTime;
        duration = inDuration;
    }

    // Question 2.4
    public String getEndTime()     {
        int endHour = Integer.parseInt(startTime.substring(0,
                        startTime.indexOf(":")))
                + (int) Math.floor(duration);
        int endMinute =
                        Integer.parseInt(startTime.substring(startTime.indexOf("
                        :") + 1))
                + (int)((duration - Math.floor(duration)) * 60);

        if (endMinute == 0)    {
            return "" + endHour + ":" + endMinute + "0";
        }
        else  {
            return "" + endHour + ":" + endMinute;
        }
    }

    // Question 2.5
    public String getAMorPM()   {
        String timeOfDay = "";

        if (startTime.startsWith("09"))  {
            timeOfDay = "AM";
        }
        else {
            timeOfDay = "PM";
        }

        return timeOfDay;
    }


    // Question 2.6
    public String getExamDate() {
        return examDate;
    }

    public String getPaper() {
        return paper;
    }
```

Private
Correct Types
Appropriate names

Assign correct values to attributes

Returns "AM" or "PM" correctly

Access method headers correct
Return correct attribute value

```
    // Question 2.7
    public String toString()
    {
        return startTime + "-" + getEndTime() + " " + paper;
    }
}
```

QUESTION 3

```
// Question 3.1
public class ExamDay   {
    // Question 3.2
    private String examDate;
    private Exam morning;
    private Exam afternoon;

    // Question 3.3
    public ExamDay(  String inExamDate, Exam inMorning,
                           Exam inAfternoon)    {
        examDate = inExamDate;
        morning = inMorning;
        afternoon = inAfternoon;
    }

    // Question 3.4
    public String getExamDate()    {
        return examDate;
    }

    // Question 3.5
    public void setExam (Exam exam)    {
        if (exam.getAMorPM().equals("AM"))  {
            morning = exam;
        }
        else if (exam.getAMorPM().equals("PM")) {
            afternoon = exam;
        }
    }

    // Question 3.6
    public String toString()    {
        String returnString = examDate;

        if (morning != null)   {
            returnString += "\nAM: " + morning.toString();
        }
        else  {
            returnString += "\nAM: No Exam";
        }

        if (afternoon != null)  {
            returnString += "\nPM: " + afternoon.toString();
        }
        else  {
            returnString += "\nPM: No Exam";
        }

        return returnString;
    }
}
```

ExamDate declared correctly
Morning and afternoon declared as Exam

ExamDate assigned correctly
Afternoon and morning assigned correctly

Calls getAMorPM() and compares
Afternoon and morning assigned correctly

Checks if morning and afternoon are null
Handles "No Exam" case
Handles exam case by calling toString

QUESTIONS 4 AND 6.1

```
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;

// Question 4.1
public class ExamManager {
    // Question 4.2
    private Exam[] examsArr = new Exam[37];
    private ExamDay[] examDaysArr = new ExamDay[31];
    private int examDayCounter = 0;

    // Question 4.3
    public ExamManager(String filename)     {
        try  {
            Scanner fin = new Scanner(new FileReader(filename));
            int counter = 0;

            while (fin.hasNextLine())  {
                String line = fin.nextLine();
                String tokens[] = line.split(",");
                examsArr[counter++] = new Exam (tokens[0],
                        tokens[1], tokens[2], Double.parseDouble(tokens[3]));
            }

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

    // Question 4.4
    public String displayAll()    {
        String returnString = "";
        for (int i = 0; i < examsArr.length; i++)    {
            returnString += examsArr[i].toString() + "\n";
        }

        return returnString;
    }

    // Question 4.5
    public Exam getExamOnDay(String searchDate, String timeOfDay)
    {
        Exam found = null;
        for (int i = 0; i < examsArr.length; i++)   {
            if (examsArr[i].getExamDate().equals(searchDate)  &&
                    examsArr[i].getAMorPM().equals(timeOfDay))     {
                found = examsArr[i];
            }
        }
        return found;
    }
```

Array sizes correct

```
// Question 6.1
   public String populateTimeTable(String filename)   {
       String subjects[] = new String[6];
       try      {
           Scanner fin = new Scanner(new FileReader(filename));

           for(int i = 0; i < 6; i++)   {
               subjects[i] = fin.nextLine();
           }
       }
       catch (FileNotFoundException fne)
       {
           System.out.println("Subjects file not found");
       }

       String returnString = "";

       for(int i = 1; i <= 31; i++)   {
           String currentDate = "2017-03-";

           if (i < 10)    {
               currentDate += "0" + i;
           }
           else  {
               currentDate += i;
           }

           Exam am = null;
           Exam pm = null;

           for(int j = 0; j < subjects.length; j++) {
               if (getExamOnDay(currentDate, "AM") != null &&
                       getExamOnDay(currentDate,
                       "AM").getPaper().contains(subjects[j]))   {
                   am = getExamOnDay(currentDate, "AM");
               }
             if (getExamOnDay(currentDate, "PM") != null &&
                       getExamOnDay(currentDate,
                       "PM").getPaper().contains(subjects[j]))   {
                   pm = getExamOnDay(currentDate, "PM");
               }
           }

           examDaysArr[i-1] = new ExamDay(currentDate, am, pm);
           returnString += examDaysArr[i-1].toString() + "\n\n";
       }
       return returnString;
   }
}
```

Uses file
Stores contents in a variable or array

Generates all the days in March
Correct with zero padding

Loops and
Finds if the AM or PM exam contains one of the subjects from the file
And records the result of the search for each

QUESTIONS 5 AND 6.2

```
// Question 5.1
public class ExamUI {
    public static void main(String args[])   {
        // Question 5.2
        ExamManager eman = new ExamManager("timetable.txt");

        // Question 5.3
        System.out.println("All Exams\n=========\n" +
                    eman.displayAll());

        // Question 6.2
        System.out.println("My Exams\n=========\n" +
                    eman.populateTimeTable("mysubjects.txt"));

    }

}
```

DELPHI SOLUTION:

QUESTION 2

```
unit uExam;

interface

uses SysUtils, Math;

// Question 2.1
type TExam = class
  private
    // Question 2.2
    examDate : String;
    paper : String;
    startTime : String;
    duration : real;
  public
    Constructor Create(inExamDate, inPaper, inStartTime : String; inDuration :
real);
    function getEndtime : String;
    function getAMorPM : String;
    function getExamDate : string;
    function getPaper : String;
    function toString : String;
end;

implementation

{ Exam }

// Question 2.3
constructor TExam.Create (inExamDate, inPaper, inStartTime: String;
  inDuration: real);
begin
   examDate := inExamDate;
   paper := inPaper;
   startTime := inStartTime;
   duration := inDuration;
end;

// Question 2.4
```

Private
Correct types
Appropriate names

Assign correct values to attributes

```
function TExam.getEndtime: String;
var
  endHour, endMinute : integer;
begin
  endHour := StrToInt(Copy(startTime, 1, Pos(':', startTime) - 1)) +
Floor(duration);
  endMinute := StrToInt(Copy(startTime, Pos(':', startTime) + 1,
Length(startTime) - Pos(':', startTime) + 1)) + Trunc(duration -
Trunc(duration)) * 60;

  if (endMinute = 0) then
    begin
      Result := IntToStr(endHour) + ':' + IntToStr(endMinute) + '0';
    end
  else
    begin
      Result := IntToStr(endHour) + ':' + IntToStr(endMinute);
    end;
end;


// Question 2.5
function TExam.getAMorPM: String;
var
  timeOfDay : String;
begin
  timeOfDay := '';

  if (Copy(startTime, 1, 2) = '09') then
    begin
      timeOfDay := 'AM';
    end
  else
    begin
      timeOfDay := 'PM';
    end;

  Result := timeOfDay;
end;

// Question 2.6
function TExam.getExamDate: string;
begin
  Result := examDate;
end;

function TExam.getPaper: String;
begin
  Result := paper;
end;

// Question 2.7
function TExam.toString: String;
begin
  Result := startTime + '-' + getEndTime() + ' ' + paper;
end;

end.
```

Returns "AM" or "PM" correctly

Access method headers correct
Return correct attribute value

QUESTION 3

```
// Question 3.1
unit uExamDay;

interface

uses uExam;

type TExamDay = class
  private
    // Question 3.2
    examDate : String;
    morning : TExam;
    afternoon : TExam;
  public
    Constructor Create(inExamDate : String; inMorning, inAfternoon : TExam);
    function getExamDate : String;
    procedure setExam(exam : TExam);
    function toString : String;
  end;
implementation

{ TExamDay }

// Question 3.3
constructor TExamDay.Create(inExamDate: String; inMorning,
  inAfternoon: TExam);
begin
  examDate := inExamDate;
  morning := inMorning;
  afternoon := inAfternoon;
end;

// Question 3.4
function TExamDay.getExamDate: String;
begin
  Result := examDate;
end;

// Question 3.5
procedure TExamDay.setExam (exam: TExam);
begin
  if (exam.getAMorPM() = 'AM') then
    begin
      morning := exam;
    end
  else if (exam.getAMorPM() = 'PM') then
    begin
      afternoon := exam;
    end;
end;
```

ExamDate declared correctly
Morning and afternoon declared as Exam

ExamDate assigned correctly
Afternoon and morning assigned correctly

Calls getAMorPM() and compares
Afternoon and morning assigned correctly

```
// Question 3.6
function TExamDay.toString: String;
begin
  Result := examDate;

  if (morning <> nil) then
    begin
      Result := Result + #10#13 + 'AM : ' + morning.toString();
    end
  else
    begin
      Result := Result + #10#13 + 'AM: No Exam';
    end;

  if (afternoon <> nil) then
    begin
      Result := Result + #10#13 + 'PM : ' + afternoon.toString();
    end
  else
    begin
      Result := Result + #10#13 + 'PM: No Exam';
    end;

end;

end.
```

Checks if morning and afternoon are null
Handles "No Exam" case
Handles exam case by calling toString

QUESTIONS 4 AND 6.1

```
unit uExamManager;
interface
uses SysUtils, Dialogs, uExam, uExamDay;
// Question 4.1
type TExamManager = class
  private
    // Question 4.2
    examsArr : array[1..37] of TExam;
    examDaysArr : array[1..31] of TExamDay;
    examDayCounter : integer;
  public
    Constructor Create(filename : String);
    function displayAll : String;
    function getExamOnDay(searchDate, timeOfDay : String) : TExam;
    function populateTimeTable(filename : String) : String;
end;
implementation
// Question 4.3
constructor TExamManager.Create(filename: String);
var
  infile : TextFile;
  line : String;
  count : integer;
  examDate, paper, startTime : String;
  duration : double;
begin
  examDayCounter := 0;

  If (FileExists(filename)) then
    begin
      AssignFile(infile, filename);
      Reset(infile);
```

Array sizes correct

```
      count := 0;

      while (NOT(EOF(infile))) do
        begin
          ReadLn(infile, line);

          examDate := Copy(line, 1, Pos(',', line) - 1);
          Delete(line, 1, Pos(',', line));

          paper := Copy(line, 1, Pos(',', line) - 1);
          Delete(line, 1, Pos(',', line));

          startTime := Copy(line, 1, Pos(',', line) - 1);
          Delete(line, 1, Pos(',', line));

          duration := StrToFloat(line);

          Inc(count);
          examsArr[count]  := TExam.Create (examDate, paper, startTime,
                       duration);
        end;

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

// Question 4.4
function TExamManager.displayAll: String;
var
  loop : integer;
begin
  Result := '';

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

// Question 4.5
function TExamManager.getExamOnDay(searchDate, timeOfDay: String): TExam;
var
  found : TExam;
  loop : integer;
begin
  found := nil;

  for loop := 1 to Length(examsArr) do
    begin
      if  ((examsArr[loop].getExamDate() = searchDate)  AND
          (examsArr[loop].getAMorPM() =  timeOfDay))  then
          begin
            found := examsArr[loop];
          end;
    end;

  Result := found;
end;


// Question 6.1
```

```
function TExamManager.populateTimeTable(filename: String): String;
var
  subjects : array[1..6] of String;
  infile : TextFile;
  line : String;
  currentDate : String;
  am, pm : TExam;
  outer, inner, count : integer;
begin
  if (FileExists(filename)) then
    begin
      AssignFile(infile, filename);
      Reset(infile);
      count := 0;
      while (NOT(EOF(infile))) do
        begin
          Inc(count);
          ReadLn(infile, subjects[count]);
        end;

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

    Result := '';
    for outer := 1 to 31 do
      begin
        currentDate := '2017-03-';
        if (outer < 10) then
          begin
            currentDate := currentDate + '0' + IntToStr(outer);
          end
        else
          begin
            currentDate := currentDate + IntToStr(outer);
          end;
        am := nil;
        pm := nil;
        for inner := 1 to Length(subjects) do
          begin
            if (getExamOnDay(currentDate, 'AM'
            and (Pos(subjects[inner],
                     getExamOnDay(currentDate,'AM').getPaper()) > 0) then
              begin
                am := getExamOnDay(currentDate, 'AM');
              end;

            if (getExamOnDay(currentDate, 'PM') <> nil)
            and (Pos(subjects[inner],
                     getExamOnDay(currentDate,'PM').getPaper()) > 0) then
              begin
                pm := getExamOnDay(currentDate, 'PM');
              end;
          end;
        examDaysArr[outer] := TExamDay.Create(currentDate, am, pm);
        Result := Result + examDaysArr[outer].toString + #10#13 + #10#13;
      end;
   end;
end.
```

Uses file
Stores contents in a variable or array

Generates all the days in March
Correct with zero padding

Loops and
Finds if the AM or PM exam contains one of the subjects from the file
And records the result of the search for each

QUESTIONS 5 AND 6.2

```
// Question 5.1
program ExamUI;

{$APPTYPE CONSOLE}

{$R *.res}

uses
  System.SysUtils,
  uExam in 'uExam.pas',
  uExamDay in 'uExamDay.pas',
  uExamManager in 'uExamManager.pas';
var
  temp: string ;
  eman : TExamManager;
begin
  try
    // Question 5.2
    eman := TExamManager.Create('timetable.txt');

    // Question 5.3
    Writeln('All Exams' + #10#13 + '=========' + #10#13 + eman.displayAll);


    // Question 6.2
    Writeln('My Exams' + #10#13 + '========' + #10#13 +
      eman.populateTimeTable('mysubjects.txt'));

    Readln(temp);
  except
    on E: Exception do
      Writeln(E.ClassName, ': ', E.Message);
  end;
end.
```

OUTPUT

SECTION A STRUCTURED QUERY LANGUAGE

QUESTION 1.1

Paper	StartTime	EndTime
Afrikaans First Add Language PI (Reading)	14:00	16:30
Afrikaans First Add Language PII (Writing)	14:00	16:30
English First Add Language PI (Reading)	14:00	16:30
English First Add Language PII (Writing)	14:00	16:30
IsiZulu First Add Language PI (Reading)	14:00	16:30
IsiZulu First Add Language PII (Writing)	14:00	16:30

QUESTION 1.2

Query1_2				
PaperID	ExamDate	Paper	StartTime	EndTime
PHY2	2017-03-03	Physical Sciences PII (Chemistry)	09:00	12:00
HIS2	2017-03-06	History PII	09:00	11:00
AFRH2	2017-03-08	Afrikaans Home Language PII (Writing)	14:00	17:00
AFRA2	2017-03-09	Afrikaans First Add Language PII (Writing)	14:00	16:30
ZUL2	2017-03-10	IsiZulu First Add Language PII (Writing)	14:00	16:30
ENGH2	2017-03-13	English Home Language PII (Writing)	14:00	17:00
MAT2	2017-03-13	Mathematics PII	09:00	12:00
ENGA2	2017-03-14	English First Add Language PII (Writing)	14:00	16:30
MATL2	2017-03-14	Mathematical Literacy PII	09:00	12:00
BUS2	2017-03-15	Business Studies PII	14:00	16:00
GEO2	2017-03-16	Geography PII	14:00	15:30
ACC2	2017-03-17	Accounting PII	14:00	16:00
CAT2	2017-03-28	Computer Applications Technology PII (Practical)	09:00	12:00
EGD2	2017-03-29	Engineering Graphics & Design PII	09:00	12:00
ITN2	2017-03-30	Information Technology PII (Practical)	09:00	12:00
LIF2	2017-03-31	Life Sciences PII	09:00	11:00

QUESTION 1.3

Query1_3		
Firstname	**Surname**	**DateOfBirth**
Ira	Bell	1999-03-07
Rigel	Blackburn	1999-03-20
Forrest	Bryan	1999-03-27
Chancellor	Buck	1999-03-29
Erich	Carey	2000-03-16
Gannon	Hays	1999-03-13
Chase	Hutchinson	1999-03-07
Ivor	Levy	2000-03-16

QUESTION 1.4

Query1_4		
Centre	**Gender**	**Centre**
1000	F	4
1000	M	2
1001	F	2
1001	M	1
1002	M	1
1003	F	3
1003	M	1
1004	M	4
1005	F	5
1005	M	3
1006	F	8
1006	M	2
1007	F	4
1007	M	4
1008	F	4
1008	M	4
1009	F	3
1009	M	5
1010	F	1
1010	M	1

Query1_4		
Centre	**Gender**	**Centre**
1011	F	2
1011	M	3
1012	F	4
1012	M	2
1013	F	2
1014	F	4
1014	M	3
1015	F	2
1015	M	4
1016	F	2
1016	M	4
1017	F	5
1017	M	3
1018	F	1
1018	M	2
1019	F	3
1019	M	1
1020	F	2
1020	M	9

QUESTION 1.5

(No output)

QUESTION 1.6

Query1_5	
PaperID	**StudentCount**
MAT1	11
MATL2	9

QUESTION 1.7

Query1_7
PaperID
ENGA1

QUESTION 1.8

Query1_8	
FullName	**LastExam**
Abbot Kemp	2017-03-03
Adria Hartman	2017-03-14
Ahmed Elliott	2017-03-23
Aladdin Haley	2017-03-24
Alden Bell	2017-03-06
Alfreda Bell	2017-03-22
Alice Herring	2017-03-09
Alisa Higgins	2017-03-14
Alyssa Joyner	2017-03-10
Alyssa Mann	2017-03-22
Andrew Faulkner	2017-03-20
Astra Fitzgerald	2017-03-13
Barbara Chan	2017-03-02
Beatrice Bradshaw	2017-03-10
Berk Hines	2017-03-02
Bethany Chase	2017-03-17
Brady Hurst	2017-03-01
Brenda Haynes	2017-03-29
Britanni Fields	2017-03-30
Callum Daugherty	2017-03-15
Cameron Garcia	2017-03-23
Carter Christian	2017-03-21
Castor Brown	2017-03-20
Cedric Burgess	2017-03-31
Chadwick Fletcher	2017-03-09
Chancellor Buck	2017-03-03
Chase Hutchinson	2017-03-20

Query1_8	
FullName	**LastExam**
Christopher Carter	2017-03-31
Clio Forbes	2017-03-02
Colette Baldwin	2017-03-24
Cora Holmes	2017-03-15
Cullen Knapp	2017-03-15
Cynthia Andrews	2017-03-10
Deacon Hays	2017-03-02
Deborah Blackburn	2017-03-29
Deborah Cain	2017-03-23
Demetria Combs	2017-03-24
Denise Kinney	2017-03-31
Desirae Gentry	2017-03-13
Dylan Lucas	2017-03-28
Eagan Alvarado	2017-03-20
Eagan Clark	2017-03-15
Eaton King	2017-03-23
Emerald Gilmore	2017-03-06
Erich Carey	2017-03-27
Eugenia Henderson	2017-03-20
Ezekiel Collier	2017-03-23
Fay Hurst	2017-03-14
Forrest Bryan	2017-03-16
Forrest Burch	2017-03-27
Fuller Fisher	2017-03-24
Fulton Bauer	2017-03-31
Gannon Hays	2017-03-14
Garrison Gallegos	2017-03-31

Query1_8	
FullName	**LastExam**
Grace Guthrie	2017-03-13
Hakeem Juarez	2017-03-16
Hamish Brewer	2017-03-16
Hanna Delacruz	2017-03-21
Harding Colon	2017-03-10
Hasad Garcia	2017-03-14
Hayden Daniel	2017-03-22
Hector Beasley	2017-03-23
Ian Gaines	2017-03-23
Imogene Holden	2017-03-30
Ira Bell	2017-03-23
Ivan Brady	2017-03-08
Ivor Levy	2017-03-30
Jackson Gillespie	2017-03-09
Jael Clay	2017-03-21
Jameson Carrillo	2017-03-24
Jasper Hinton	2017-03-31
Jemima Burton	2017-03-23
Jesse Lloyd	2017-03-23
Julie Espinoza	2017-03-30
Justine Fox	2017-03-08
Karyn Delacruz	2017-03-17
Kathleen Donovan	2017-03-17
Keelie Boyd	2017-03-28
Keiko Abbott	2017-03-31
Kelly Henson	2017-03-28
Kelsie Atkinson	2017-03-08
Kennedy Hernandez	2017-03-17
Kessie Cline	2017-03-14
Laurel Flynn	2017-03-17
Lee Jefferson	2017-03-15
Lila Grant	2017-03-13
Lila Manning	2017-03-01

Query1_8	
FullName	**LastExam**
Gisela Langley	2017-03-03
Lois Leon	2017-03-17
Lynn Becker	2017-03-20
Maggy Harding	2017-03-23
Maxwell Hester	2017-03-17
May Brock	2017-03-01
Melyssa Gillespie	2017-03-20
Nichole Avila	2017-03-28
Nina Chan	2017-03-31
Odette Day	2017-03-16
Oleg Herman	2017-03-13
Olympia Dillon	2017-03-13
Oren Blanchard	2017-03-23
Paki Bridges	2017-03-10
Pandora Green	2017-03-15
Priscilla Glover	2017-03-27
Quail Erickson	2017-03-09
Rae Giles	2017-03-10
Rahim Edwards	2017-03-10
Rigel Blackburn	2017-03-15
Roanna Madden	2017-03-07
Sarah Cooke	2017-03-30
Shea Hunt	2017-03-24
Slade Logan	2017-03-23
Solomon Cline	2017-03-22
Sybill Ferrell	2017-03-28
Tate Bates	2017-03-16
Thor Cleveland	2017-03-09
Uriel Chavez	2017-03-24
Vincent Berg	2017-03-15
Walter Duncan	2017-03-17
Wanda Blair	2017-03-21
Whoopi Diaz	2017-03-27

SECTION B OBJECT-ORIENTED PROGRAMMING

FINAL OUTPUT:

```
All Exams
=========
09:00-12:00 Economics
14:00-17:00 Afrikaans Home Language PI (Reading)
09:00-12:00 Physical Sciences PI (Physics)
14:00-16:30 Afrikaans First Add Language PI (Reading)
09:00-12:00 Physical Sciences PII (Chemistry)
14:00-16:30 IsiZulu First Add Language PI (Reading)
09:00-11:00 History PII
14:00-17:00 English Home Language PI (Reading)
09:00-12:00 Mathematics PI
14:00-16:30 English First Add Language PI (Reading)
09:00-12:00 Mathematical Literacy PI
14:00-17:00 Afrikaans Home Language PII (Writing)
09:00-12:00 Visual Arts PI
14:00-16:30 Afrikaans First Add Language PII (Writing)
09:00-12:00 Consumer Studies
14:00-16:30 IsiZulu First Add Language PII (Writing)
09:00-12:00 Mathematics PII
14:00-17:00 English Home Language PII (Writing)
09:00-12:00 Mathematical Literacy PII
14:00-16:30 English First Add Language PII (Writing)
09:00-12:00 Life Sciences PI
14:00-16:00 Business Studies PII
09:00-12:00 Tourism
14:00-15:30 Geography PII
09:00-11:00 Business Studies PI
14:00-16:00 Accounting PII
09:00-12:00 Information Technology PI (Theory)
14:00-17:00 History PI
09:00-12:00 Computer Applications Technology PI (Theory)
09:00-12:00 Dramatic Arts
09:00-12:00 Engineering Graphics & Design PI
09:00-12:00 Geography PI
09:00-11:00 Accounting PI
09:00-12:00 Computer Applications Technology PII (Practical)
09:00-12:00 Engineering Graphics & Design PII
09:00-12:00 Information Technology PII (Practical)
09:00-11:00 Life Sciences PII

My Exams
=========
2017-03-01
AM: No Exam
PM: No Exam

2017-03-02
AM: 09:00-12:00 Physical Sciences PI (Physics)
PM: 14:00-16:30 Afrikaans First Add Language PI (Reading)

2017-03-03
AM: 09:00-12:00 Physical Sciences PII (Chemistry)
PM: No Exam
```

```
2017-03-04
AM: No Exam
PM: No Exam

2017-03-05
AM: No Exam
PM: No Exam

2017-03-06
AM: No Exam
PM: 14:00-17:00 English Home Language PI (Reading)

2017-03-07
AM: 09:00-12:00 Mathematics PI
PM: No Exam

2017-03-08
AM: No Exam
PM: No Exam

2017-03-09
AM: No Exam
PM: 14:00-16:30 Afrikaans First Add Language PII (Writing)

2017-03-10
AM: No Exam
PM: No Exam

2017-03-11
AM: No Exam
PM: No Exam

2017-03-12
AM: No Exam
PM: No Exam

2017-03-13
AM: 09:00-12:00 Mathematics PII
PM: 14:00-17:00 English Home Language PII (Writing)

2017-03-14
AM: No Exam
PM: No Exam

2017-03-15
AM: No Exam
PM: No Exam

2017-03-16
AM: No Exam
PM: 14:00-15:30 Geography PII

2017-03-17
AM: No Exam
PM: No Exam
```

2017-03-18
AM: No Exam
PM: No Exam

2017-03-19
AM: No Exam
PM: No Exam

2017-03-20
AM: 09:00-12:00 Information Technology PI (Theory)
PM: No Exam

2017-03-21
AM: No Exam
PM: No Exam

2017-03-22
AM: No Exam
PM: No Exam

2017-03-23
AM: No Exam
PM: No Exam

2017-03-24
AM: 09:00-12:00 Geography PI
PM: No Exam

2017-03-25
AM: No Exam
PM: No Exam

2017-03-26
AM: No Exam
PM: No Exam

2017-03-27
AM: No Exam
PM: No Exam

2017-03-28
AM: No Exam
PM: No Exam

2017-03-29
AM: No Exam
PM: No Exam

2017-03-30
AM: 09:00-12:00 Information Technology PII (Practical)
PM: No Exam

2017-03-31
AM: No Exam
PM: No Exam