basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

INFORMATION TECHNOLOGY

EXAMINATION GUIDELINES

GRADE 12

2014

These guidelines consist of 13 pages and 4 annexures.

TABLE OF CONTENTS

1. INTRODUCTION

The Curriculum and Assessment Policy Statement (CAPS) for Information Technology outlines the nature and purpose of the subject Information Technology. This guides the philosophy underlying the teaching and assessment of the subject in Grade 12.

The purpose of these Examination Guidelines is to:

- Provide clarity on the depth and scope of the content to be assessed in the Grade 12 National Senior Certificate Examination in Information Technology.
- Assist teachers to adequately prepare learners for the examinations.

This document deals with the final Grade 12 external examinations. It does not deal in any depth with the School-Based Assessment (SBA), Practical Assessment Tasks (PATs) or final external practical examinations, as these are clarified in a separate PAT document which is updated annually.

These Examination Guidelines should be read in conjunction with:

- The *National Curriculum Statement (NCS) Curriculum and Assessment Policy Statement (CAPS)*: *Information Technology*
- The National Protocol of Assessment: *An addendum to the policy document, the National Senior Certificate: A qualification at Level 4 on the National Qualifications Framework (NQF), regarding the National Protocol for Assessment (Grades R–12)*
- The national policy pertaining to the programme and promotion requirements of the National Curriculum Statement, Grades R–12

2.   COMPUTER LAB REQUIREMENTS

Refer to the CAPS document for the resources required for offering Information Technology, with respect to the infrastructure, equipment and finances that are the responsibility of the school.

The two approved high-level programming languages for Information Technology are Delphi and Java. Schools must make sure that the required **programming language is installed** on all computers **at the beginning of the year**. This is to ensure that Grade 12 candidates have enough time to familiarise themselves with the software they are required to use during the final Information Technology (IT) practical examination.

2.1   Software requirements

The following software is the minimum required for examination purposes in Information Technology. The versions of software are the requirements for the 2014 Grade 12 practical IT examination session (Paper 1). However, these versions will be revised/confirmed annually.

2.1.1 Delphi

Learner files will be prepared using Delphi 7.0. Any database files that may be required will be prepared using MSAccess 2007 (mdb format). These files are all upwards compatible.

NOTE: • The '*Microsoft Jet 4.0 OLEDB Provider*' driver can be used.
            • If you prefer to use the '*Microsoft Office 12.0 Access Database Engine OLE DB Driver*', the Microsoft Office Access database engine 2010 (English) version must be installed instead of the '*Microsoft Jet 4.0 OLEDB Provider*' driver.

2.1.2 Java

The following software **must be installed** and used at schools using Java as programming language.

NOTE: The specific versions of the three software tools must be exactly as specified in the list below. If a school uses different versions, the Department of Basic Education (DBE) will not be held responsible for problems that may occur during the Grade 12 final practical examination session due to software problems.

Java software required:

•   NetBeans 7.3.1
•   Java JDK 1.7 for 32-bit computer systems (Update 25)
•   Java DB 10.8.2.2. This is supplied with JDK 1.7 and **must** be installed.

NOTE:
- The configuration of Java software mentioned on the previous page will be used by the DBE to prepare the learner files for the practical question paper. The computer lab that will be used during the Grade 12 practical examination session must be prepared using the same software configuration.
- Teachers must ensure that a **DerbyEmbedded library** is created on each computer for learners to prepare for the practical examination AND for learners to be able to do their PAT AND to answer possible database questions during the practical examination. Follow the steps provided in Part 1 of ANNEXURE C on how to create a DerbyEmbedded library.

2.2 Official checklist for the practical examination (Paper 1)

According to examination instruction Circular E14 of 2013, each computer centre where Grade 12 candidates will do their final IT practical examination must be audited. The auditing must be done using the checklist (ANNEXURE A) which is supplied annually by the DBE/provincial district offices. The certification for readiness (ANNEXURE A of Circular E14) must be submitted to the DBE after being co-signed by the IT teacher, the IT technician and the school principal, and verified by the district facilitator.

3. ASSESSMENT IN GRADE 12

3.1 Format of question papers

PAPER	TYPE OF PAPER	DURATION	TOTAL	MARKING
1	**Practical** Structured questions	3 hours	150	External
2	**Theory** Structured questions	3 hours	150	External

Questions in both Papers 1 and 2 will assess performance at different cognitive levels with an emphasis on the application of knowledge, critical thinking skills and problem-solving techniques. Learners are required to be able to investigate and analyse problems in a variety of contexts (such as scientific, technological, environmental and everyday-life contexts) in order to solve the described problems effectively, either via programming code in Paper 1 or describe proposed solutions in Paper 1/Paper 2.

3.1.1 Paper 1: Practical paper

This will be a practically-oriented question paper containing questions on software solution development. This question paper assesses **practical skills** as well as the knowledge and understanding underlying programming skills pertaining to software solution development making use of the high-level programming language studied (Delphi/Java Netbeans). Software development skills using Scratch/BYOB is excluded.

The following topics should be prepared/could be examined in the practical question paper:

- Sound programming principles
- Use of data of all data types specified in the IT CAPS document
- Basic programming statements specified in the IT CAPS document, such as the selection structures (e.g. IF statement) and repetition structures (e.g. FOR statement)
- All GUI components mentioned in both the IT CAPS document and this document. These are the minimum required components. Learners are encouraged to explore other components and to use more than the minimum required. Refer to the table (ANNEXURE B) to see how the different components will be referred to in the question paper. Since the question paper is not programming language specific, the general terms listed in the table will be used to refer to components.
- Data structures, such as arrays
- Object-oriented programming (OOP)
- Databases and manipulation of data without using SQL. Manipulation of data in a database using data-aware components as stated in the IT CAPS document

NOTE: SQL-statements will be assessed in the theory paper (Paper 2).

- Character handling, which includes string manipulation and the use of text files
- All other structures and utility classes mentioned in the IT CAPS document

(a) Physical requirements

To successfully complete the practical paper:

- Each learner must have access to his or her own computer in the examination room.
- The software specified in Section 2, Computer lab requirements, of this document must be installed on the computers that will be used during the examination session **well in advance** of the examination date.

(b) Layout of Paper 1

Paper 1 will assess the following broad topics:

	MARKS
SECTION A	**45 (± 10)**
Basic programming skills with standard/basic structures listed in the IT CAPS document	
SECTION B	**70 (± 10)**
• Database manipulation	
• Object-oriented programming (OOP)	
NOTE: SECTION B can contain questions on one or both topics listed in this section.	
SECTION C	**35 (± 10)**
General problem-solving – open-ended question	
	150

GUI (included in all questions)

- GUIs will be provided in most cases, however, for simple/basic programs where only a form/frame and a few components are required, learners could be asked to create the GUI themselves.
- Learners are expected to be able to add/delete/modify components of a GUI provided.
- Learners must be able to dynamically instantiate components.

3.1.2 Paper 2: Theory paper

The question paper will include all theory aspects of all content, concepts and skills of topics, as well as elements of software solution development, e.g. algorithm development, data structures, program design and general programming concepts as well as generic problem-solving questions.

Questions will cover work from Grades 10 to 12 as indicated in the IT CAPS document.

Layout of Paper 2

The following format will be used based on the format specified in the IT CAPS document:

SECTION			DESCRIPTION
A	Human Computer Interaction and Social Implications	These topics could be integrated as part of the other sections and will not be a separate section in the question paper.	**Short Questions**(±10 marks) A range of short questions covering all topics that could include multiple-choice and modified true/false items.
B			**Systems Technologies** (±20 marks) Questions related to the content, concepts and skills in the Systems Technologies topic.
C			**Communications Technologies and Network Technologies** (±30 marks) Questions related to the content, concepts and skills in the Communication Technologies and Network Technologies topic. (Including the Internet)
D			**Data and Information Management** (±20 marks) Questions related to the management of data and the concept of information management.
E			**Solution Development** (±30 marks) Questions aligned to the Solution Development topic which assesses the knowledge and understanding underlying the concepts and skills in the Solution Development topic. Software design tools for examination purposes as part of the theory paper are limited to basic flow charts, class diagrams and use case diagrams (UCDs). Write SQL code for database manipulation.
F			**Integrated Scenario** (±40 marks) This section is based on a scenario and assesses topics from all sections (B to E).

3.2   Cognitive levels

Papers 1 and 2 will include questions across three cognitive levels. The distribution of cognitive levels in the practical and theory papers is given in the table below.

Cognitive level	Description	Paper 1 (Practical)	Paper 2 (Theory)
1	Knowledge and remembering	30%	30%
2	Understanding and applying	40%	40%
3	Analysing, Evaluating and Creating	30%	30%

4.   ELABORATION OF THE CONTENT FOR GRADE 12 (CAPS)

4.1   Paper 1: Practical examination

4.1.1 Possible types of questions per section

Learners must always apply good programming principles and skills as well as sound programming techniques. Learners could be required to apply/use the following concepts in all sections of the question paper:

- Basic programming skills listed in SECTION A below
- Variables, primitive and complex data types
- Control statements (decision and repetition statements)

NOTE: Although topics are classified per section below, this classification will not always be followed to the letter. Some of the concepts could appear in other sections than listed below. Therefore, this is only an indication of what learners can expect in the practical question paper.

SECTION A

Basic programming skills with standard/basic algorithms studied and listed in the CAPS

Examples of tasks learners must be able to code/perform (not limited to this list only):

- Swap values
- Determine/Calculate the highest, lowest, average, mean, percentage increase/ decrease, etc. of a number of values
- Basic calculations, such as calculating area, volume, VAT, discount
- Determine whether a number is even or odd, prime, composite, etc.
- Determine whether a number is a factor/multiple of another number
- Isolate digits in an integer number
- Lowest common multiple (LCM), greatest common divisor (GCD)
- Determine age based on a given date
- Explore algorithms for general string manipulation, e.g. use an ID number to determine age and gender, count vowels, identify palindromes
- Basic input, processing and validation techniques, e.g. test for division by zero, test for entries outside allowed boundaries, test for no input

- Constructs/Controls:
  o Decision-making: IF, Case/Switch
  o Iteration: Conditional and unconditional loops
- Convert pseudocode/flow chart representation of a program/program segment to a functional program
- Modify/Correct a program/program segment

SECTION B

Database manipulation with problem-solving

Learners must be able to manipulate a normalised relational database using data-aware components (ANNEXURE B) and code constructs from Delphi/Java Netbeans (not SQL) to do the following:

- Perform CRUD operations (create/add, read, update, delete) records
- Deletion of records in a relational database will only be required to perform on one table (the 'many' table in a one-to-many relationship – the table that contains the foreign key).
- Manipulate data retrieved from the database, e.g. calculate VAT on prices stored in the database and create a report with VAT inclusive prices. Text manipulation, e.g. creating passwords from fields, such as name, surname, ID.
- Solve problems using data retrieved from the database that could include algorithms as listed in the IT CAPS document
- Do queries on the database/extract records meeting specific condition(s)

NOTE:
- A normalised relational database which contains a ***minimum of two tables*** will be provided.
- Learners will not be required to change the structure of the database or to enter large quantities of data via the keyboard.
- A folder that contains the project/program with a successful connection to the database will be supplied.
- If the database connection is NOT successful, learners must be able to do the connection themselves. Instructions on how to do the connection to the database successfully have been provided as part of ANNEXURES C (Java) and D (Delphi).
- Text file(s) will be supplied to be able to create the database if the database supplied is cannot be used.
- **Outline of the content of ANNEXURE C: JAVA**

Part 1: How to create a DerbyEmbedded library
NOTE: Teachers must perform the tasks listed in Part 1 of ANNEXURE C to prepare the examination centre prior to the practical examination.

Part 2: A guideline that lists steps on how to prepare a common database connection which will assist the teacher for assessment purposes. This will eliminate the need to reconnect when project folders are transferred between computers.

Part 3: Steps on how to create a database connection if the database connection that was supplied was unsuccessful

- **Outline of the content of ANNEXURE D: DELPHI**

The connection to the database that will be supplied can either be done with or without using a Data Module unit. Learners must be familiar with both ways of setting up the connection string.

NOTE: The program supplied by the DBE for this question will make use of the Data Module.

Part 1: How to import text files into an MSAccess database

Part 2: Steps on how to establish a database connection using a Data Module

Part 3: Steps on how to establish a database connection without using a Data Module

Object-orientated programming (OOP)

This question entails the design and development of solutions that require the application of OOP principles. Learners must be able use the following OOP concepts effectively and correctly to solve problems in the best possible way within the OOP paradigm:

- Classes and objects
- Parameterised and non-parameterised constructors
- Private and public attributes and methods
- toString method
- Accessor, mutator and auxiliary methods

Examples of types of questions:

- Write methods to expand given classes.
- Write a complete class.
- Correct incorrect code contained in a method.
- Improve the structure of a given but poorly designed class.
- Instantiate objects.
- Call methods of the object to perform tasks/display information.

NOTE: The programming skills listed as part of SECTIONS A and C could also be required to be applied in the OOP question.

SECTION C

General problem-solving – open-ended question

The questions tested in this section will be of an open-ended nature where the learner must be able to apply all the programming techniques and skills studied during Grades 10, 11 and 12 in Information Technology to solve the problem.

Design and develop solutions using first principles to solve problems that can require learners to do the following:

- Apply any of the basic algorithms listed as part of basic programming skills (SECTION A).
- Search with/without a flag.
- Sort alphabetically and numerically.
- Determine highest/lowest/average/total in a 1D/2D array.
- Remove/Add elements from/to a 1D/2D array/lists or database table.
- Edit/Update elements in a 1D/2D array/lists or database table.
- Display all/selected information.
- Validate input/defensive programming techniques to prevent incorrect user input.
- Use text file as a means of input and/or output.
- Do character handling.
- Use parameter passing between procedures/functions/methods.
- Use custom/any required algorithms to solve the problem.
- Make use of multiple forms/frames (Tabsheets are not regarded as Forms).
- Create dynamic GUI components.

NOTE:
- Learners can use any programming constructs/data structures studied to solve the problem, including concepts listed in SECTIONS A and B, such as database manipulation and OOP solutions.
- Learners could be required to do code from first principles (using algorithms) to search, sort and read/write from/ to a file instead of using pre-written built-in features such as .sort and .locate.

The emphasis in IT is to gain programming skills and therefore to develop code using first principles as far as possible. That is to ensure that learners know the logic behind some of the supplied built-in functions.

4.1.2 Excluded for Paper 1

The following concepts will not be examinable in the practical paper, but learners may make use of these concepts/constructs in their PAT:

- Array of objects; arrays as an attribute of an object
- Inheritance and polymorphism applied in OOP-programming
- Database manipulation using SQL-statements (SQL will be tested in the theory paper)
- Software design techniques, methods and tools, such as UML, CRC cards

4.1.3 Learner files for Paper 1: Practical paper

Learner files will be supplied and may include:

- GUI(s)
- Text file(s)
- Database
- Object class

4.2 Paper 2: Theory examination

4.2.1 Latest trends in IT technology

Approximately 10 marks will be allocated to questions on the latest developments in Information Technology. A topic that will be focussed on will be announced annually.

TOPIC FOR 2014:

Wearable electronic devices, such as the Google glasses

The following aspects on the announced topic will be assessed:

- Explanation/definition
- How does the technology work? A general explanation is required. The learner must show an understanding of how the technology works. Diagrams (if applicable) can be asked including explanatory notes.

NOTE: Lengthy extremely technical and detailed explanations are not required. For example, if the technology involves the use of Bluetooth technology, it will not be required to explain the technical detail of how Bluetooth works.

- Applications: Who can benefit from using the technology?
- Possible ethical issues

4.2.2 Solution Development (SECTION E)

None of the questions in SECTION E (Solution Development) will be specific to a programming language. Learners may be required to write out the solution of a given problem in terms of an algorithm.

Questions from past question papers can be used as a guideline to the type of questions that will be asked and includes the following:

- Explain solution development concepts, e.g. referential integrity, OOP, human-computer interaction (HCI)
- Solution development tools:
  o Analyse/draw a class diagram
  o Analyse/draw a trace table
  o Analyse/draw a simple flow diagram
  o Analyse/compile pseudocode to solve a simple problem
- Recommend programming structure/functions/statements that would be best suited to solve a problem
- Explain/apply/answer questions on the concept such as:
  o Parameter passing
  o Arrays
  o Handling exceptions
  o Data validation and data integrity
  o Human-computer interaction (HCI)
  o GUI

- Learners could be required to write out some SQL statements. Theoretical questions on SQL will be formulated similar to SQL questions from previous practical IT question papers. The following SQL statements could be assessed and is applicable to one table or more than one linked table (relational database):

  o   Select, distinct
  o   Where
  o   Order by
  o   Group by
  o   Special operators: Between, In, Like, Is Null, Having
  o   Insert, Update, Delete
  o   Subqueries (simple form, single select or aggregate)
  o   Aggregate functions (sum, average, min, max, count)
  o   Date functions (day, month, year, date)
  o   String functions (length, left, right, mid, concatenation of fields)
  o   Type casting of fields
  o   Dynamic queries making use of user input
  o   Queries with parameters where a user input is given to modify data in a table or to search a table
  o   Calculating new fields

4.2.3 General guidelines to complete Paper 2

- The mark allocation per question indicates the number of facts required.
- Avoid general answers, such as 'cheap', 'fast'.
- If asked to list two facts and the learner supplies more than two facts, only the first two facts supplied will be marked.
- Do not leave answers blank. Always try to formulate an answer.

5.   CONCLUSION

It is envisaged that this Examination Guidelines document will serve as an instrument to strengthen and empower teachers to set valid and reliable assessment items in all their classroom activities.

This Examination Guidelines document is meant to articulate the assessment aspirations espoused in the CAPS document. It is therefore not a substitute for the CAPS document which teachers should teach to.

Qualitative curriculum coverage as enunciated in the CAPS cannot be over-emphasised.

ANNEXURE A: Official checklist for the practical examination (Paper 1)

CERTIFICATION (State of Examination Readiness)

REQUIRED FOR THE CONDUCT OF THE FINAL PRACTICAL EXAMINATION IN COMPUTER APPLICATIONS TECHNOLOGY AND INFORMATION TECHNOLOGY – NATIONAL SENIOR CERTIFICATE (NSC) EXAMINATIONS: GRADE 12

(To be completed by the PED at least ONE month before the examination)

School name		Tel. No	
District		Fax No.	
EMIS No.		Cell No.	
Examination centre No.			
E-mail address			
Name of examination officer		Cell No.	
Name(s) of IT teacher(s)	1.	2.	
Name(s) of CAT teacher(s)	1.	2.	
Name of IT technician who will be on standby during the examination		Cell No.	

1	How many Grade 12 learners will be writing CAT in 2014?	
2	How many Grade 12 learners will be writing IT in 2014?	
3	How many operational computers are available for one examination sitting?	
4	How many Grade 12 CAT classes does your school have?	
5	How many Grade 12 IT classes does your school have?	
6	How many computer centres does your school have?	
7	How many examination sittings will be required? One or two?	
8	Will your learners be writing at a venue other than your own school examination centre? If so, please state the venue: _____	
9	What plans are in place in the case of a power failure?	
10	Are the computers in the centre networked?	
11	If the computers are networked, what type of network environment is installed? E.g. peer-to-peer; client-server	
12	Which operating system is in use at your school examination centre? E.g. Windows 7	
13	Which software application suite is in use? E.g. Microsoft Office 2010 or OpenOffice 4.1	
14	Indicate with a tick √ or cross × which of the following programmes is/are properly installed on the computers in the examination centre: Word processor ☐ Spreadsheet ☐ Database ☐ Presentations ☐	
15	Have the latest service packs been installed? Refer to bullet 4 and 5 of Circular E of 2013. Latest service packs will also be appropriate.	
16	Have the regional settings been set to South Africa, e.g. Date and time and the units of measurement in the word processing application set to centimetres?	
17	Have the computers been set to save at 5 minute intervals for every application?	

18	Will accessories such as Paint be installed and available for the candidates?	
19	Which antivirus program is installed?	
20	At what intervals is the antivirus program updated?	
21	Has the Recycle Bin on every computer been emptied/Will the Recycle Bin be emptied one day before the examination?	
22	Is there a plan in place for emptying the Recycle Bin on every computer before the 2nd sitting?	
23	Has every computer been cleared of all other files/folders/documents not needed for the examination? (Work of other classes must be isolated.)	
24	Does the school have CD/DVD writers installed?	
25	Has the feature for using a flash memory disk by the learners been disabled on all computers?	
26	What option will be used to transfer the learners' work onto the final DVD/CD to be submitted for marking? E.g. flash memory disk; network	
27	Are printers installed on each computer being used for the CAT examination? (It is necessary to install and test a printer on each computer, even though printing may or may not take place.)	
28	What type of printers is available for printing during the IT examination? E.g. laser	
29	Are security systems in place to prevent the computers from being stolen or accessed?	
30	Does the school have a maintenance plan/budget in place for the computer centre?	

CAT TEACHER	IT TEACHER
_____ NAME	_____ NAME
_____ SIGNATURE	_____ SIGNATURE
_____ DATE	_____ DATE

SCHOOL PRINCIPAL	IT TECHNICIAN
_____ NAME	_____ NAME
_____ SIGNATURE	_____ SIGNATURE
_____ DATE	_____ DATE

SCHOOL STAMP

Verified by

WARD MANAGER/SUBJECT ADVISOR/DISTRICT EXAMINATIONS & ASSESSMENT SERVICES/ DISTRICT ICT SERVICES	Comments, if any
_____ NAME	_____ _____
_____ SIGNATURE	_____ _____
_____ DATE	

ANNEXURE B: GUI components (Paper 1)

The table below indicates the list of components learners will be required to use. The question paper will be set in programming terms that are not specific to any particular programming language and therefore it will refer to the different GUI components as follows:

Component	Delphi	Java	Reference in the question paper
Form	TForm	JFrame	Form
	TPageControl	TabbedPane	Tabbed form/sheets
Button	TButton TBitButton	JButton	Button
Label	TLabel	JLabel	Label
Panel	TPanel	JPanel	Panel
Radio group	TRadioGroup	ButtonGroup	Radio group
Text box	TEdit	JTextField	Text box
	TRichEdit	JTextArea	Output area
Combo box	TComboBox	JComboBox	Combo box
List box	TListBox	JList	List box
Check box	TCheckBox	JCheckBox	Check box
Radio button	TRadioButton	JRadioButton	Radio button
Spinner	TSpinEdit	JSpinner	Spinner
Grids/Tables	TStringGrid	JTable	Grid
	TDBGrid TADOTable TADOQuery TDBGrid		
Menu options	TMainMenu	JMenuBar JMenuItem	Menu options component
'Runtime' components			
Message box	InputBox ShowMessage MessageDlg	JOptionPane. showInputDialog	Message dialog
Other components	TDataSource		

NOTE: Components that are used regularly and learners are familiar with, e.g. Button, Text box, Output area and Panel, may be required to be instantiated dynamically.

ANNEXURE C: JAVA – Methods to establish database connectivity

Transferring a database project from one computer to another without reconnecting to the database:

PART 1

Software Requirements (as mentioned before):
- NetBeans 7.3.1
- Java JDK1.7u25
- Java DB installation set to location
  - C:\Program Files\java\JDK1.7.0_25\db
  - Do not install the Glassfish server as it has a different Java DB version.
- Create the following library to ensure that the Derby-jar files are available:
  - Step 1: Select Tools → Ant Libraries → Ant Library Manager do the following:
  - Step 2: Create a new Library called **DerbyEmbedded**.
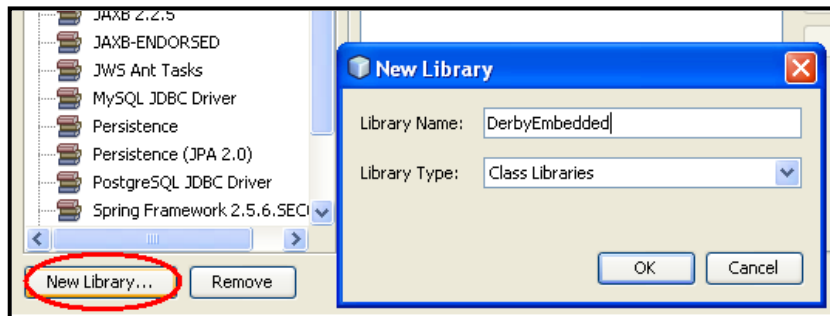
  - Step 3: Select Add JAR/Folder.
  - Step 4: Browse and select the Derby.jar file in the db/lib folder of the JDK installation.
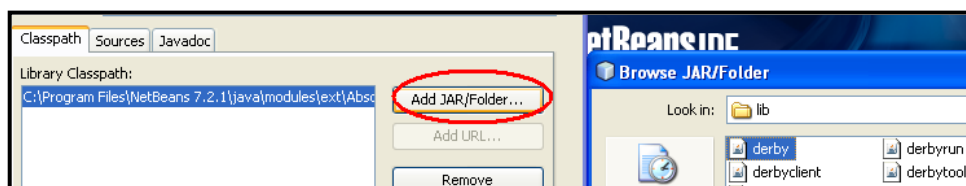
  - Step 5: Click OK.

PART 2

If the initial database project is set up using the following steps, the project may be used at other computers without learners having to reconnect manually.

This way of connecting eliminates the need for the learners to do the following themselves:

- Start the server in the services window.
- Change the database location in Java DB properties window.
- Enter a username and password.
- Resolve the missing Derbyclient.jar problem.

> If you want to see the database in the services window, then follow the first three bullets.

The following example program will take you through the DB connectivity using the Embedded database setup.
NOTE:
- Only one connection is allowed.
- No server is needed.

Example:
The database called **ExampleDatabase** has been created in the NetBeans Services Window and saved in a folder called **ExampleDB**. The steps illustrated below will ensure that the database project will connect on all computers (most suitably used for assessment purposes).

Step 1   Create a NetBeans Java Application project called **Example**.
Step 2   Create a Package called **ExamplePackage** in the **Example** project.
Step 3   Create a JFrame called **ExampleGUI** in the **ExamplePackage**.
Step 4   Copy the **ExampleDB** folder into the project root directory.
Step 5   Create Entity Classes.

                Right click on **ExamplePackage**.
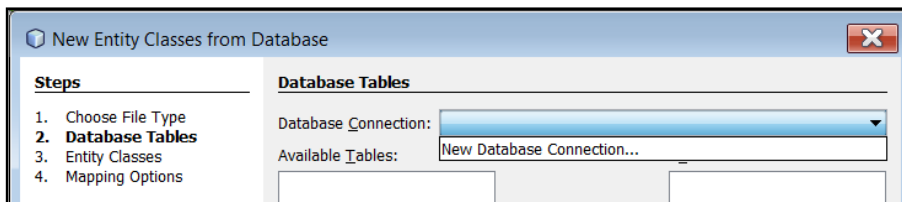   Select   New → Entity Classes from Database.
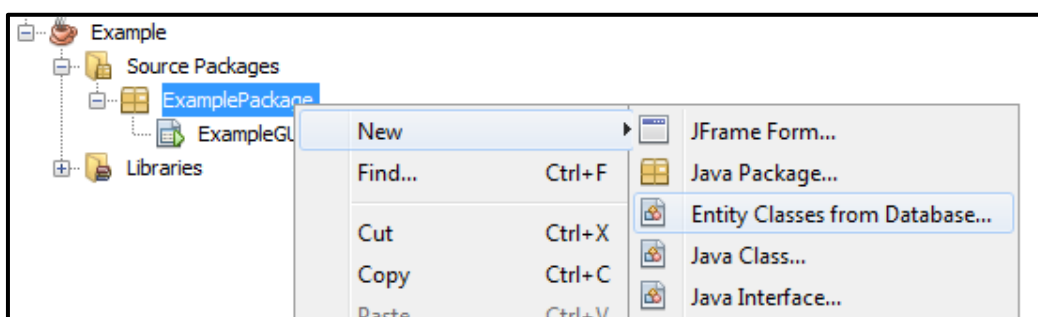
If the Entity Classes from Database is not a menu option:
- Choose Other…
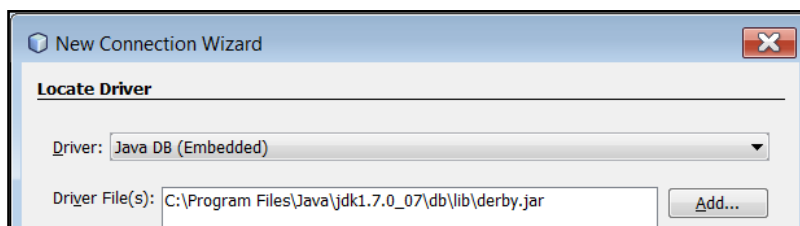- Select the class from the File Types list.
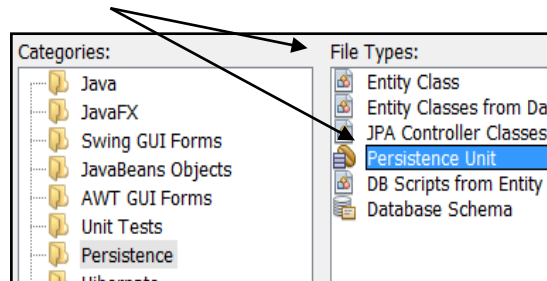
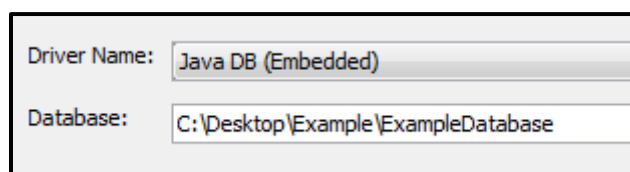In the New Entity from Database Dialog:
- Select New Database Connection.

- Select Java DB (Embedded) and click Next.

- On the **New Connection Wizard** next to Database:
 Enter the full path to **ExampleDB**.

- User Name: nbuser
- Password: nbuser
- Click on Test Connection.


NOTE: We did not touch the services window.

Now do the following:
- Click →Next →Next and Finish to the New Entity Classes from Database dialog.
- Move **ExampleTable** to the Selected list.
- Uncheck Include related tables (there is only one table). Click → Next
- Keep only Create Persistence Unit checked. Click→Next
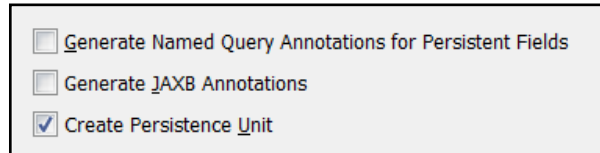
- Collection Type: java.util.List and click → Finish

The result is the creation of both the entity class Exampletable and the Persistence Unit as persistence.xml in the META-INF folder.
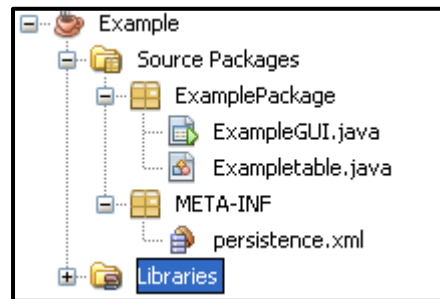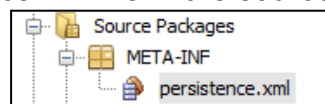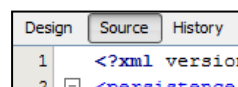

Very Important

NOTE:
- In the services window the connection node is indicated and connected.
- The embedded driver can only manage one connection at a time.
- To make this connection work on any machine, the absolute path must be removed.
- This is how it is done:

    o Click on the persistence.xml file in the source package

    o and in its source

    o find the URL to the database.
    o Delete the part   C:/ … /Example/ . Save the change.

The connection to ExampleDatabase is still active. When we run the project, it will try to create a second connection and throw an error.

The ExampleDatabase must be disconnected in the services window.

- Right click in the Libraries node of the project window and select Add library.
- Select the DerbyEmbedded library.

At this point your database connection is set up.

Add a JTable onto ExampleGUI.
- Right Click →Select Bind→Elements→Import Data to Form
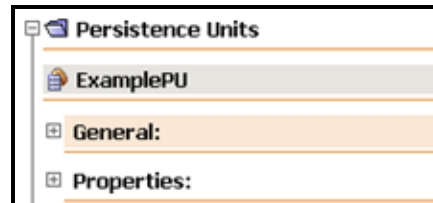- Select the available connection; i.e. the one in the services window.
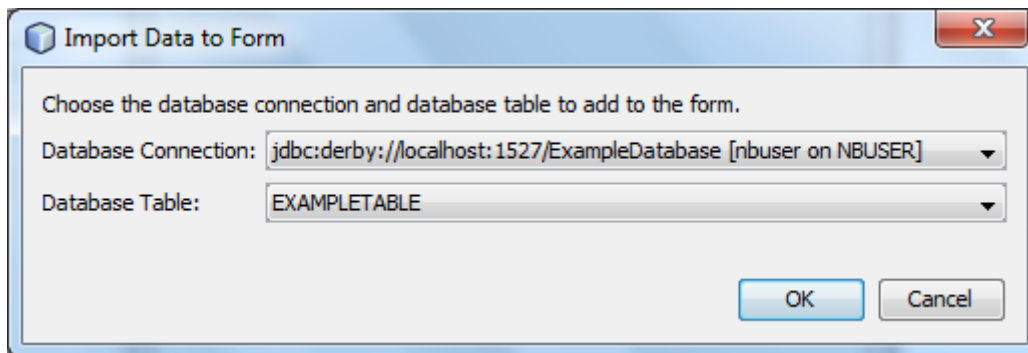
    ExampleTable should appear as the **Database table**.
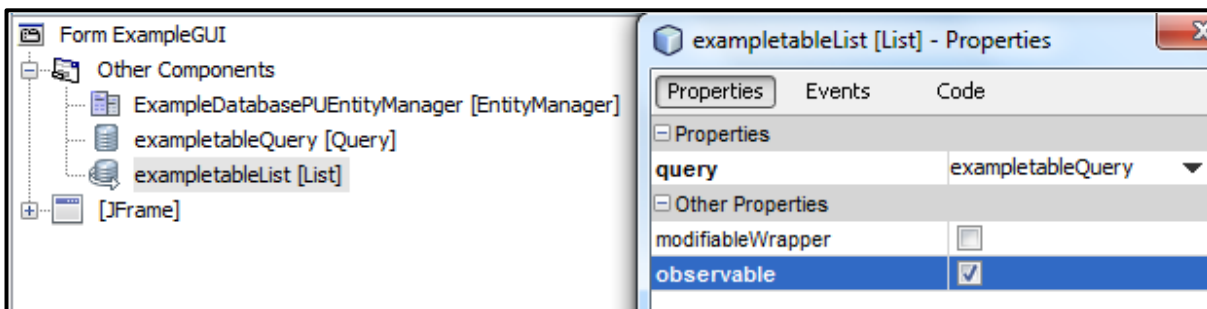- Click → OK→OK

Note the persistence objects in the Navigator window.

- Check the persistence.xml file – Design tab.
- Scroll down to see a second Persistence Unit.
- Remove the duplicate unit to leave only the correct Persistence Unit (ExamplePU).

Steps to make the list observable >

- Right click in the jTable in design view
- Open Navigator → Select Properties → Tick Observable

Disconnect the connection in the services window.

PART 3

A text file can be used to create the table(s) and/or relationship. A text file with the necessary information will also be supplied in the examination that can be used if the connection did not function properly.

In the services window:
- Create database (browse to project folder)
- Right click -> Connect
- NBUSER -> Right click on tables
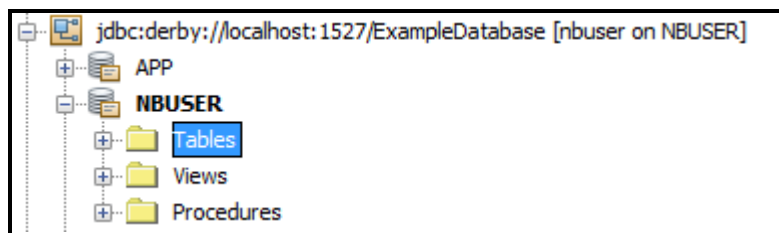
- Execute Command.
- Open file -> Given file with extension .sql
- Run.
- Choose correct connection.

Example of file:

```
CREATETABLEtblAthletes (
        AthleteID INTEGER NOTNULL,
        FirstNameVARCHAR (25) NOTNULL,
        SurnameVARCHAR (35),
        Gender VARCHAR (1) NOTNULL,
        Age INTEGER NOTNULL,
        SchoolNameVARCHAR (35) NOTNULL,
        PRIMARYKEY (AthleteID)
);

INSERTINTONBUSER.TBLATHLETES (ATHLETEID, FIRSTNAME, SURNAME, GENDER, AGE, SCHOOLNAME)
        VALUES (1001, 'Robert ', 'Bee ', 'M', 14, 'Birtley High School');
INSERTINTONBUSER.TBLATHLETES (ATHLETEID, FIRSTNAME, SURNAME, GENDER, AGE, SCHOOLNAME)
        VALUES (1002, 'Scott ', 'Holmes ', 'M', 14, 'Birtley High School');
INSERTINTONBUSER.TBLATHLETES (ATHLETEID, FIRSTNAME, SURNAME, GENDER, AGE, SCHOOLNAME)
        VALUES (1003, 'Nathan ', 'Williams ', 'M', 14, 'Birtley High School');
INSERTINTONBUSER.TBLATHLETES (ATHLETEID, FIRSTNAME, SURNAME, GENDER, AGE, SCHOOLNAME)
        VALUES (1004, 'Paul ', 'Vitterson ', 'M', 14, 'Birtley High School');
INSERTINTONBUSER.TBLATHLETES (ATHLETEID, FIRSTNAME, SURNAME, GENDER, AGE, SCHOOLNAME)
        VALUES (1005, 'Roger ', 'Shooter ', 'M', 14, 'Birtley High School');
```

ANNEXURE D: DELPHI – Methods to establish database connectivity

PART 1

The database as well as text files which contain the contents of the database tables will be provided. However, if for some unexpected reason the database file cannot be used, learners must be able to create their own database using the provided text files. This can be done as follows in MSAccess 2010:

- Open MSAccess.
- Create a blank database or open the given database.
- Select **External Data| Text File| Browse** to the text file | Click on **OK**.
- Follow the steps the Wizard provides to import the provided text file as a table.
- Study the structure and content of the table you require (it should be provided in the question paper) and select the options the Wizard provides accordingly.
- The first line in the text files contains the field names to be used. Make sure you select this option (checkbox) in the Wizard.
- Select: *Choose my own primary key* since the data provided normally already contains a primary key field.
- Once the table has been created and the content or field names are incorrect, simply delete the table, import the file again and make sure you select the correct options in the Wizard.
- If there is more than one text file, repeat the process for the remainder of the text file(s).
- Always make sure that the database is in the same folder as your Delphi program.

PART 2 (Using a data module)

Set up a link to a Database in Delphi using the Data Module (Recommended)

Example:
Assume you are working with a database called **DBSchool** which contains a table named **tblTeachers**.

Do the following to set up the Database connection:

1. Create a new folder for the program, e.g. **SchoolInfo**. Store/Copy the *Access* database called **DBSchool** into this folder.

2. Create a new Delphi application. Save the unit and the project in the folder that contains the database (**SchoolInfo**).
    - Save As ... *frmSchool_u.pas*.
    - Save Project As ... *frmSchool _p.dpr*.

3. Create a Data Module as follows:
    In Delphi 7:   Select **File** | **New** | Data Module
    In Delphi 2010: Select **File** | **New** | Other | Delphi Files | **Data Module**
    - Change the name of the Data Module in the Object Inspector to *dmSchool*.
    - Save the Data Module as *dmSchool_u.pas* in the same folder as the project and unit of the Form: **File** | **Save As**.

4. Place an ADOConnection component on the Data Module.
    (In Delphi 7: from the ADO page, In Delphi 2010: from the dbGo menu).
    o   Change the name of the ADOConnection to conSchool.
    o   Click on the three dots (ellipses) next to the *ConnectionString* property.
    o   Click on **Build**.
    o   Choose the driver you are using (*Microsoft Jet 4.0 OLE DB Provider/ Microsoft Office 12.0 Access Database Engine OLE DB Provider*).
    o   Click on **Next**.
    o   Click on the three dots (ellipses), and select the name of the database.
    o   Click on the **Advanced** tab, deselect the default option **Share Deny None** and select the **Read/Write** option.
    o   Click **OK** and click **OK** again to close the ConnectionString dialog.
    o   In the Object Inspector, set the LoginPrompt property to False.
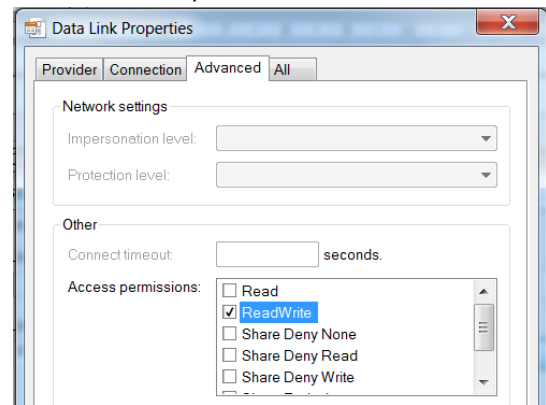
5. Place an **ADOTable** component (from the ADO page (Delphi 7) | dbGo menu (Delphi 2010)) on the Data Module.
    o   Change the name to *tblTeachers*.
    o   Change the Connection property to *conSchool.*
    o   Change the TableName property to *Teachers*.

6. Place a **DataSource** component (from the Data Access page/menu) on the Data Module.
    o   Change the name to *dsrTeachers*.
    o   Change the DataSet property to *tblTeachers*.

7. To link the Form to the Data Module:
    o   Go to the code of the Unit of the Form (*frmSchool_u*).
    o   Add the name of the Data Module to the uses clause.

```
unit frmSchool_u;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, dmSchool_u;
```

8. Place a **DBGrid** (from the Data Controls page/menu) on the Form. Name it *dbgTeachers*.
   o Change the DataSource property to *dmSchool.dsrTeachers*.

9. Select the **ADOTable** component (*tblTeachers*) on the Data Module.
   o Change the Active property to True.

10. Display the Form. The contents of Teachers table will now be displayed in the DBGrid.

11. Click **File | Save All** to ensure all changes made to the Form and the Data Module is saved.

PART 3 (Without using a Data Module)

Set up a link to a Database in Delphi without using the Data Module

Open your Delphi application/Create a new Delphi application and do the following:

- Place an **ADOTable** (from the ADO page), a **DataSource** (from the Data Access page) and a **DBGrid** (from the Data Controls page) on the Form. Name the components properly e.g. tblTeachers, dsrSchool and dbgTeachers.
- Click on the **ADOTable** component and select the ConnectionString property.
   o Click on the ellipsis button (three dots) to the right of the ConnectionString property in the Object Inspector.
   o Click on Build, which opens the Data Link Properties dialogue window.
   o Click on the Provider tab and select *Microsoft Jet 4.0 OLE DB Provider/ Microsoft Office 12.0 Access Database Engine OLE DB Provider*. Click on Next.
   o Browse and look for the database file. Select the file and then click on Open.
   o Remove the username Admin.
   o Click on Test Connection.
   o Click on **OK** on each of the open dialogue windows.
   o Set the TableName property to the table you want to link to in the database.
   o Set the Active property to True.
- Click on the **DataSource** component and set the DataSet property to the name of the table.
- Click on the **DBGrid** and set the DataSource property to the specified data source component.