basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

INFORMATION TECHNOLOGY

GUIDELINES FOR PRACTICAL ASSESSMENT TASK (PAT)

Grade 11

2022

These guidelines consist of 30 pages.

Contents

What is the PAT? 3
Mark allocation 5
The Topic 6
Malpractice 8
Non-compliance 9
PAT requirements 10
Instructions 11
Annexure A 18
Annexure B 34
Learner declaration – Phase ____ 34
Annexure C 35
Declaration of authenticity 35

What is the PAT?

The Practical Assessment Task (PAT) is a software development project in which you will have the opportunity to demonstrate your software development and programming skills.

The purpose of the PAT is to:

- Work extensively with content knowledge to improve your programming and organisational skills,
- Implement computational thinking, other higher order thinking skills and formulate strategies and to solve problems on different levels,
- Develop good working practices to prepare you for the real world, such as -
    - Time management.
    - Thorough planning.
    - Perseverence to achieve and to excel in what you set out in your plan.
    - Presentation and marketing of your product.

You will need to demonstrate knowledge and understanding of the software development life cycle through analysis, design, coding and testing of your project. You will have to show effective use of the software design tools and techniques which you have studied.

The PAT is divided into **TEN TASKS** as named below:

Task no.	Task description	Stage of system development
Task 0	Problem definition and research	Requirements
Task 1	Task definition and user story Acceptance test	Requirements
Task 2	Database design	Requirements
Task 3	Navigation / flow between screens GUI design	Requirements
Task 4	Data dictionary Arrays User defined methods(optional) Text files	Design
Task 5	IPO table and data validation	Design
Task 6	Create GUI application	Design
Task 7	Create DB and connect to Delphi	Implementation
Task 8	Coding	Implementation
Task 9	Testing and data validation	Implementation
Task 10	Documentation	Review
	Interview	Evaluation

LEARNERS NEED TO STRICTLY ADHERE TO THE DUE DATES

NOTE:

Submission dates: Specific dates will be determined by your subject educator.

TASK 0: Does not carry any marks; Preparation for PAT and the research thereof.

TASK 1-5: Not later than the end of Term 3

TASK 6-10: Not later than the three weeks before examination in Term 4.

NOTE:

You will be required to demonstrate and discuss your application during an interview session

Mark allocation

The PAT counts 25% of your final examination mark for Information Technology. It is therefore crucial that you strive to produce work of a high standard.

Tasks	Task description	Maximum Mark
Task 1	Task definition and user story Acceptance test	15
Task 2	Database design	12
Task 3	Navigation / flow between screens GUI design	8
Task 4	Data dictionary Arrays User defined methods(optional) Text files	12
Task 5	IPO table and data validation	15
Task 6	Create GUI application	9
Task 7	Create DB and connect to Delphi	24
Task 8	Coding	12
Task 9	Testing and data validation	8
Task 10	Documentation	8
	Interview	7
	Total:	**130**

NOTE:

- The PAT mark is a compulsory component of the final certification mark for all candidates registered for Information Technology.
- Your PAT will be moderated at district and provincial level by subject experts.

Topic

Tournaments / Competitions for a Sport of your Choice

Crown the winner in a tournament

There are many types of activities where learners are competing at school level. These competitions range from sport, to cultural and various extra mural activities. For this year, learners are expected to design and develop a Delphi program for an interactive "**tournament style planner and updater"**

A tournament is a competition held among different teams in a particular game or activity according to a fixed schedule where a winner is eventually decided.

There are several formats used in various levels of competition in activities and games to determine an overall champion. Some of the most common are the single elimination, the best-of-series, the total points series more commonly known as on aggregate, or the round-robin tournament.

Other examples of different types of tournaments are (not limited to):

- Knock-out or Elimination Tournament (Single Knock-out or Single Elimination, Consolation Type I and Type II, C Double Knock-out or Double Elimination)
- League or Round Robin Tournament (Single League, and Double League)
- Combination Tournament (Knock-out cum Knock-out, Knock-out cum League, League cum Knock-out, League cum League)
- Challenge Tournament (Ladder and Pyramid)

Different types of tournaments have their own merits and demerits.

This project allows you to create a programme using Delphi. Use good programming principles throughout.

Make use of the following pre-scripts for the application:

- select a tournament of your choice.
- choose ONE types of tournament structures/styles for the current planned "competition"
- allow different teams/applicants to be entered.
- provide the correct tournament plan, populated with the participants/teams
- update after each match/session.
- correctly present the leaders and winners

Instructions should be clear so that anybody could use the program

- Expected additional features:
  - o The expected time to complete the tournament
  - o The expected number of officials needed to complete the event

This tournament software program is targeted at the organiser of the event.

The application needs to make use of arrays and a database.

The tournament could be based on an existing school tournament or it could be a new idea for a tournament.

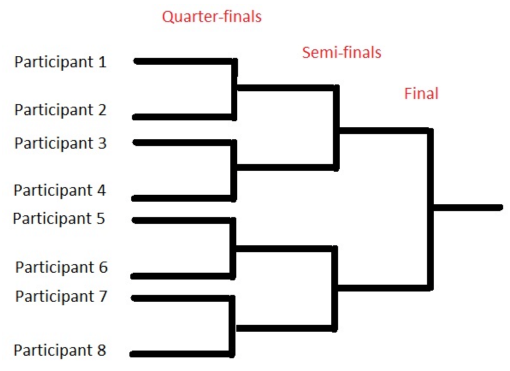
The complexity of the tournament is not awarded any marks.

Your final program must comprise ***one*** single, logically related piece of software.

Resources that could provide some background:

https://ncert.nic.in/textbook/pdf/kehp110.pdf
https://www.printyourbrackets.com/types-of-tournaments.html
https://slideplayer.com/slide/13690959/
https://www.youtube.com/watch?v=pXZV9bKO3h0

Simple example:



What you need to be able to do the PAT

To be able to do the PAT, you need the following:

- The Delphi IDE (Integrated Development Environment)
- Word processing and database software.
- Storage media to save and backup your work electronically.

Malpractice

As the PAT is an individual project that is part of your final promotion mark, you may NOT:

- Get help from others without acknowledgement.
- Allow others to do programming code for you.
- Submit work which is not your own.
- Share your work with other learners.
- Include work directly copied from books, the Internet or other sources without acknowledgement.

The above actions constitute malpractice, for which a penalty will be applied, depending on the seriousness of the offence.

Non-compliance

You will be given up to a part of term four to submit outstanding work or present yourself for the PAT. Should you fail to fulfil the Practical Assessment Task requirements, you will be awarded a zero ('0') for the PAT component of IT. This will result in an incomplete promotion mark, and it may result in you not passing your grade.

PAT requirements

The project must include the following:

- A database connection and database manipulation that entail performing different CRUD (Create, Read, Update and Delete) operations.
- A multi-form GUI with good functionality and usability, based on sound HCI principles
- The use of a text file for input/output purposes, for example to populate data structures and to provide reports.
- Other data structures eg. Arrays, that will be relevant to your program

Database

The database must:

- have at least TWO tables
- contains sufficient data volumes and uses a variety of field types
- be accessed and manipulated by the program using code constructs

GUI

The graphical user interface (GUI) must

- have at least TWO forms/screens that allows for navigation between forms depending on the user choices
- interact with the database and other data studtures to provide the necessary input, processing and output
- comply with relevant HCI principles

Text files

Your application need to make use of a text file(s) for input and/or output.

Arrays

The appropriate use of array(s).

NOTE:

The mark obtained for your project will be greatly influenced by the quality of the programming code that manipulates the data successfully to adhere to the user requirements in the best possible way. Quantity cannot replace variety, effectiveness, and quality.

Instructions

During these tasks you must show that you have done a proper and thorough user requirement analysis and design. This needs to be done to determine who the users are and what the users of the system would require it to do.

TASK 0: Research

Task 0 is a discussion and preparation task that does not carry any marks and has no deliverables.

- **Topic** is the tournament style planner.
- **Purpose of program** Describe the purpose of your program – why does the organiser need your program. Research a few different tournament styles and write down the positives of each tournament.
- **Possible solution** What will the program do to meet these needs above. Describe how your program will work. Include a description of the program and how the user will interact with your program.
- **Scope** (Explain what your program cannot do)

During this phase you have to show that you have done a proper and thorough user requirement analysis. This needs to be done to determine who the users are and what the users of the system would require it to do. The following can be used as a guideline:

TASK 1A: Define the Task

Write a brief description (approximately 200 words) in your own words to describe, in general terms, the problem/task and how the project will solve the problem.

Your explanation must highlight that:

- You understand the needs of the task that you have chosen.
- Your solution will solve the needs of the task.
- Provide a simple/ brief description of the scope of the project.

TASK 1B User story and acceptance test

The ***user*** is the target audience and will determine the needs and requirements of the program. Determine the user needs and processing requirements.

The aim is to identify the user(s), user needs and processing requirements of the system. Use a table or a 'use case diagram' to explain the role and activity of each user of the system.

For example:

As a learner **I want to** register **so that** I can play the Maths game to improve my skills

WHO	*WHAT*	*WHY*
As a …	*I want to …*	*So that …*
User/Actor/role	*Goal/program feature required*	*Value or benefit*

Use cases (goals only) – use the user stories, identify the goals that represent a functionality (functional requirement) that can be used or performed in isolation, i.e. a user will be able to request only that service/function in a single session and state in the format, (Register Learner (named by verb))

Note: Each use case could possibly represent a form/screen.

TASK 2 Design the Database

The aim is to design a database to serve as a data source as well as to manipulate data contained in the database using programming code.

Show the design of the database, including the tables, field names, field types and field sizes.
The database should provide data to the program to be processed and create reports.

The Delphi program must be able to manipulate the content of database tables, for example update/edit/delete/add data, provide results of queries, provide reports, et cetera.

TASK 3 Navigation/Description of Flow Diagram

Clearly indicate the logical program flow and navigation between screens/forms. Use a flow diagram or any other form of illustration to present a global overview of the project/system.

TASK 4 Data Dictionary

Text file(s)

Your application must use a text file(s) for input and/or output. Explain where a text file can be used in your program so that it adds value to the program.

Array(s)

Your application must use an array(s). Explain where an array can be used in your program so that it adds value to the program.

Other data structures such as user defined methods (optional)

TASK 5 Input, Processing, Output (IPO) and data validation

Use an IPO illustration/table to:

- Design the overall solution, considering all constituent parts and the interrelationships between the various parts of the program/system.
- Specify the format, data types, source of input, validation of input and error checking mechanisms.
- Specify the format, data types, source of output.
- Specify processing that needs to be done and provide algorithm(s)/formulae to show how the processing will be done.
- Provide a clear description to indicate the input, processing and output requirements of the system for at least TWO of the main interfaces.

TASK 6 Create the GUI application

Developing the GUI according to the planning document that was developed during Task 1-5. Use appropriate components to ensure easy use and effective navigation. Follow HCI principles to ensure that the application is user friendly and provides all necessary requirements for the user(s) to use the program effectively and navigate through the options/functionalities easily.

The aim is to produce a GUI design that considers good human-computer-interface (HCI) principles. Your design should include measures that prevents errors occurring due to invalid input and that minimises the amount of information a user must enter.

Use HCI design principles and design a GUI that considers the following:

- The user, type of user and context of user.
- User requirements, usability.
- Dialogues – must be relevant, simple and clear.
- Icon usage and presentation – well selected and relevant, well placed and purposely used.
- Colour – appropriate use of and combination of colours.
- Feedback – neat, clear and well presented.
- Helpful error messages.
- Exits – clearly marked, placed correctly.
- Shortcuts.
- Flow of information on the screen – top to bottom and left to right.
- Sensible use of space on the screen.

Provide examples of planned data capture and data entry designs (screen dumps/shot/grabs may be used from a prototype of the project but must be annotated) and of planned output design.

Show the GUI design following HCI principles of interface(s), excluding introductory screens.

TASK 7 Create a Database and connect to application

Design and construct the database according to the planning document that was developed during Task 1 - 5. Apply appropriate techniques and sound database development rules.

Pay attention to the following:

- Table names should start with a prefix “tbl” for example *tblSuppliers*.
- The use of spaces in field names might affect reading data from fields into the Delphi application.
- The size of text fields must be restricted/limited as the columns in the DBGrid in the Delphi application will be affected by the field size.
- The data types of fields must be well thought out as this information will ultimately connect to components in the Delphi application, for example, the difference between the Number and AutoNumber data types, the difference between saving a date as text or as a DateTime data type, etcetera.
- Keep the purpose of the project in mind when setting up fields and tables.
- Ensure that the database connects correctly to the program and interacts with the program in a meaningful and effective way that supports the program once you have written the Delphi code.

TASK 8 Write the Code

Write code to develop the program/system according to the planning document that was developed during Task 1-7. Note the following:

- Use good programming techniques and structures.
- Implement effective algorithms and sound defensive programming techniques to produce a robust program.
- Use appropriate structures to satisfy the requirements of the algorithms.
- Use nested loops and conditional structures.
- The following data structures are compulsory in addition to the database:
    - o Text file - reading OR writing OR appending
    - o Array(s)
- The use of any other data structure not already tested/ advanced programming construct
- Use relevant validation procedures and components.
- Develop a well designed and user friendly GUI
- Rename relevant components to add to readability and documentation of your code.

- Input data using the most effective method, for example a text file, database, keyboard, components
- Process the data using the most appropriate methods.
- Generate output of data using the correct components and structures, with formatting where needed.
- Ensure smooth interaction between forms/tabs.
- Correctly manipulate and query the database.

TASK 9 Testing and data validation

Test the program/system using clearly defined normal/expected data, erroneous data, and boundary (extreme) data.

Marks will be allocated for the coding of coding of validation and error catching.

Guidelines:

*“**What*** must be tested? ***Why*** must it be tested? ***When*** will it be tested? ***How*** will it be tested?

Use the use case scenarios (generally the additional scenarios) to derive test cases. Test cases must be executable.

Provide suitable test inputs, e.g. test data (normal/expected data, erroneous data and boundary data)

Provide expected results for normal data, erroneous data and boundary data.”

Example (from alternative scenario: *Empty required fields*)

Test case	**Input data**	**Expected Result**
Verify if ***Name field*** is filled	Text field not empty	Success
	Text field empty	Warning message Another chance to enter name

*“**Why***? To ensure the user entered all the data that the program requires

When? On completion of the Registration unit /When the submit button is clicked”

TASK 10 Document the Program

Project notes for the user:

These project notes must describe how the user should interact with the program. It can include notes on how to navigate through the program, specific requirements such as passwords and installation procedures if applicable. The notes must also describe any known bugs or problems. Project notes can be written as part of the help function of the program. Tool tip texts can also be provided.

Project notes for developers:

These project notes could include specifications/limitations applicable to the project to ensure that the program is installed and set up correctly, e.g. the connection to the database.

Project notes related to the programming code should be embedded as comments in the code. Document the code so that other programmers will be able to interpret the code and understand the purpose of individual pieces of code. It should also include comments to explain sections of complex code.

Hand In

Hand in:

- The completed Delphi project (Delphi code, text files, database and any other resources required to execute the program successfully) and project notes.
- The declaration of help received **(Annexure B).**
- The declaration of authenticity **(Annexure C).**

Interview

Demonstrate your program and answer questions about the program and the code during an interview session.

Guidelines for the demonstration of the project:

- The teacher will schedule dates and times for demonstrations. About 15 minutes per project will be allowed.
- You should hand in all the documentation before the demonstration takes place – at least one week in advance.
- The demonstrations must be done electronically on the computer.
- You must execute your computer program and show all the features of the program to the teacher for evaluation.
- The teacher can require you to execute test procedures to make sure that the entire program is working correctly.
- As part of the demonstration, the teacher will identify random pieces of programming code in the project and ask you to explain the purpose and working thereof. This is done to ensure that you did the coding yourself. A similar type of procedure will be followed during moderation. If you cannot explain the code used in the project, no marks can be awarded for the project.
- You must hand in the electronic copy of the project that was demonstrated. The teacher will use this copy to allocate any outstanding marks to finalise the mark.

Annexure A: Assessment tools

Learner Name:

Task 1A		**3**	**2**	**1**	**0**		
Task definition (Short description ±150 - 200 words)	-	Task is clearly stated and described in the learner's own words (Clear statement of the purpose and audience). Shows a thorough understanding of what the problem/task involves. Explains a possible solution to solving the problem/task.	The task is clearly stated and described in the learner's own words, but minor shortcomings. Shows a good understanding of what the problem/task involves. Covers almost all aspects.	The statement is vague, leaving the reader unsure of what the purpose of the program will be. Minimal understanding of what the problem involves. Minimal coverage of aspects.	No statement / statement is totally inadequate not applicable. Poor or no coverage of aspects.	3	
Task 1B	**4**	**3**	**2**	**1**	**0**		
Role, activity, value (who, what, why) □ Who will use the system? □ What are the goals/ activities that user will perform? □ Why do they want/need it?	Role, activity, value of all users (at least 2 different types of users) of the system thoroughly and correctly described. Well documented, clear and to the point.	Role, activity, value of all users (at least 2 different types of users) of the system described but minor shortcomings e.g. one instance where goal is not clear, value not clear, etc. Well documented, but minor shortcomings.	Many shortcomings in discussion of role, activity, value of users, e.g. two instances where goal is not clear, value not clear, etc. Only 1 type of user of the system discussed. Not well documented but still acceptable	Major shortcomings in discussion of role, activity, value of users, e.g. many parts left out or incorrect information Poorly documented – not acceptable	Not done or incorrect or irrelevant	4	
Use Case diagram or table format (Top level)	Clearly describes what the system does to accommodate all needs of all users as described by user stories. Clearly and correctly shows □ All requirements/goals □ System boundary □ All actors (at least two) □ Actors' interaction with the system	Describes what the system does to accommodate needs of users but with minor shortcomings One aspect lacking or not clearly described, e.g. no system boundary, not all goals, not all actors, interaction incorrect, not according to user stories, etc.	Describes what the system does to accommodate needs of users but shortcomings. Two aspects lacking or not clearly described	Describes what the system does in some instances. Major shortcomings Three aspects lacking or not clearly described	More than three aspects lacking or no use case diagram or totally incorrect	4	

List of functions the program should perform	Detailed list which clearly and correctly defines at least 5 functions All functions derived from the user requirements	Fairly detailed list which clearly and correctly defines 4 functions At least 3 functions derived from the user requirements	A list which defines 3 functions containing some detail Only 2 functions derived from user requirements	A list which defines 2 functions but lacking in detail Only 1 function derived from user requirements	No list of functions included Very rudimentary, no detail Not derived from user requirements	**4**	
						15	

Learner Name:						
Task 2	**3**	**2**	**1**	**0**		
Choice of fields	Well-chosen fields. All fields contribute to the solution Contains no field that could be calculated from other data	One field does not contribute to the solution or One field that could be calculated	More than one field do not contribute to the solution Or More than one field that could be calculated	No database or incorrect or irrelevant	**3**	
Field types and size	All fields well-chosen in terms of type and field size	One field not well-chosen in terms of type or field size	More than one field not well chosen in terms of type or size	No database	**3**	
Appropriateness – Tables	At least 2 tables. Correct use of fields for relationships between tables.	Only one table/only the fields listed – no relationships.	Only one table/only the fields listed – no relationships. No/incorrect primary key.	No database.	**3**	
Role of DB (no SQL) (***How*** DB will be manipulated, e.g. within a dataset, access fields and records, navigate records, modify individual fields and records and apply changes, etc.) Role of manipulation in program described/ motivated	Manipulation and interaction well described and well-motivated. Most appropriate to meet requirements	Manipulation and interaction not clearly described or substantiated in one instance. Adequate to meet requirements	Manipulation and interaction not well described or motivated in two instances. Mostly not meeting requirements	No database or incorrect or irrelevant or no interaction	**3**	
					12	

Learner Name:

Task 3	4	3	2	1	0		
A diagrammatical representation of the design and flow of events when the program is used	An excellent attempt to show the sequence of all steps and flow of events when the program is executed with no shortcomings	A good attempt to show the sequence of all steps and flow of events when the program is executed with minor shortcomings	A satisfactory attempt to show the sequence of steps and flow of events when the program is executed with significant shortcomings	A poor attempt to show the sequence of steps and flow of events when the program is executed with major shortcomings	No diagram OR Incorrect, irrelevant or unsuitable for the application	**4**	
HCI (look and feel) □ user interaction □ Consistency □ logical flow □ grouping □ Help/friendly dialog	Good GUI design All of the listed principles applied throughout the system.	Satisfactory GUI design Most of the principles (at least 3) applied throughout the system.	Limited GUI design Most of the principles (at least 2) applied throughout the system.	Poor GUI design Applied less than 50% (less than 2) of the principles	GUI design not functional OR Does not support the intended use at all	**4**	
						8	

Learner Name:

Task 4

Data Dictionary	4	3	2	1	0		
Variables and components □ Variety of and correct use of appropriate variable types and components □ Correct use of local and global variables □ Proper naming convention of variables for example iNumber, sName etc. □ Correct prefix for components for example edt, red, cmb etc.	Excellent – all four aspects applied correctly in all instances.	Good – one aspect omitted or not used well.	Satisfactory – two aspects omitted or not used well.	Limited – more than two aspects omitted or not used well.	Totally inappropriate or incorrectly applied	**4**	
Data Structures (Excl. Database)	**4**	**3**	**2**	**1**	**0**		
Text files(s)	Excellent and relevant use of a text file	Good use of a text file	Limited use of a text file.	An attempt to use a text file with short comings.	Not done or incorrect or irrelevant.	**4**	
Array(s)	Excellent and relevant use of an array	Good use of an array	Limited use of an array with minor shortcomings	An attempt to use an array with many short comings.	Not done or incorrect or irrelevant.	**4**	
						12	

Learner Name:

Task 5	**3**	**2**	**1**	**0**		
Input (***How*** input will be obtained and managed)	Clearly describes all inputs in terms of how input has to be obtained, how sources of input will be used and the format of the input	Minor shortcomings in descriptions. One or two inputs not clearly described.	Limited description. More than two inputs not clearly described.	Input requirements not described; incorrect or irrelevant	3	
Processing (What processing will be done)	Clear list of 6 processors to be done	One or two processors not listed	About 50% of processors listed	No processors listed	3	
Processing (***How*** processing will be managed)	Clearly describes all processing/ manipulation in terms of how data has to be processed/manipulated/ transformed (algorithms, formulas, etc.)	One or two processing/manipulations not clearly described.	More than two processing/ manipulations not clearly described	Processing/manipulation not described, incorrect or irrelevant	3	
Output (***How*** output will be managed) ☐ Data output ☐ Format of output eg. Date ☐ Component of output eg dbgrid	Clearly describes all outputs in terms of how it will be displayed, how sources of output will be used, as well as the format, type and size of the output	Minor shortcomings in descriptions. One or two outputs not clearly described.	Limited description. More than two outputs not clearly described.	Output requirements not described, incorrect or irrelevant	3	
Other software design tools – any one	**3**	**2**	**1**	**0**		
One of: ☐ use case diagrams ☐ class diagrams ☐ other	Excellent use of tool that clearly defines the program.	Tool constructed with shortcomings.	Tool not clearly constructed.	No tool, incorrect use of tool, irrelevant to application.	3	
					15	

Learner Name:

Task 6	**3**	**2**	**1**	**0**		
HCI principles (***How*** GUI will meet the users' needs).Does it consider: ◻ Purpose of program and user ◻ Standard GUI design principles ◻ Ease of use, logic flow ◻ Clearly marked navigation ◻ Friendly dialogue ◻ Help	Good GUI design, considering almost all (at least 5) of the principles for at least 2 of the main interfaces, excluding the introductory screen.	Satisfactory GUI design, considering most (at least 4) of the principles for at least 2 of the main interfaces, excluding the introductory screens.	Limited GUI design, considering only 50% (at least 3) of the principles for at least two of the main interfaces, excluding the introductory screens.	Poor GUI design considering less than 50% (less than 2) of the principles considered.	**3**	
Components	Most appropriate components used in all cases Excellent layout All choices clearly substantiated	In one or two cases another component would have been more appropriate Satisfactory layout One or two choices not substantiated	In three or four cases another component would have been more appropriate Layout not satisfactory Three or four choices not substantiated	Inappropriate components used in more than four cases or choices not substantiated in more than four cases	**3**	
TOE chart/ description	Clearly describes the responsibility and events for all of the components/controls	Responsibility and events for one or two of the components/controls not clearly described	Responsibility and events for more than two of the components/controls not clearly described	No TOE chart or all components/controls poorly described	**3**	
					9	

Task 7	4	3	2	1	0		
Use and interaction ***HLL: high level language**	Use of a relevant database with at least 2 tables and 1 relationship. Best suited fields, data types and sizes in all instances Well-chosen database manipulation through HLL code that contributes to the solution. Excellent, smooth interaction with HLL.	Use of a relevant database with at least 2 tables and no relationship. Mostly well-chosen fields, data types and sizes. Good manipulation through HLL code that mostly contributes to the solution. Good, smooth interaction with HLL.	Use of relevant database with 1 table. Acceptable fields, data types and sizes. Satisfactory interaction with HLL. Satisfactory manipulation trough HLL code but does not always contribute to the solution.	Use of relevant database. One table. Few suitable fields, data types and sizes. Limited manipulation trough HLL code.	No interactions. Or no manipulations. Or no database. Or totally inappropriate.	4	
Embedded database - Manipulation using HLL native operations / code constructs			**2**	**1**	**0**		
Delete record(s) and apply changes			Meaningfully, appropriately and correctly done Contributes to solution	Mostly meaningful, appropriate and correctly done Mostly contributes to solution	Not done or totally inappropriate or not meaningful	2	
Insert record(s) and apply changes						2	
Edit/Update records/ selected fields in record(s) and apply changes						2	
Validate field(s) (when record is inserted/modified)						2	
Read/View selected fields and records						2	
Navigate through records in a dataset (first, next, previous, etc.) using methods						2	
Manipulate dataset object & records, apply changes (Filter/Sort/Search records) and refresh						2	
Use event/listener that triggers after record pointer has moved						2	
At least 1 report (output displayed/text file) as a result of processing/data transformation						2	
At least 1 report (output displayed/text file)as a result of querying						2	
						24	

Learner name:							
Task 8	**4**	**3**	**2**	**1**	**0**		
Data structures (User defined, excl. DB)	A variety of data structures used correctly (arrays, text files, etc.) and appropriately	A variety of data structures used (arrays and text files) but minor shortcomings, e.g. not always correctly or appropriately used	Used at least one data structure (e.g. array or text file) correctly and appropriately	Used at least one data structure (e.g. array or text file) but not always correctly and appropriately	Totally inappropriate or incorrect or not used	**4**	
Input (database, text file, user)	All three types used. Most appropriate, effective input strategies (database, text files, user input) used in all instances.	All three types used, but in one instance a more appropriate or effective input strategy could be used	In two instances a more appropriate or effective input strategy could be used	In more than two instances a more appropriate or effective input strategy could be used	Totally inappropriate or ineffective	**4**	
Output (coded)	In all cases: Most appropriate display, well formatted/readable/structured / understandable, e.g. headings and repeated on following page/screen where applicable. No logical errors. All processing results are correct.	In all cases: Most appropriate display, well formatted/readable/structured / understandable, but with minor shortcomings in one instance One minor logical error One result problematic	In most cases Appropriate display Some logical errors Some of the results are not correct.	In some cases Appropriate display Many logical errors Difficult to read output Many results incorrect	Many logical errors. Almost all the results are incorrect/few of the required results are delivered	**4**	
						12	

Learner Name:

Task 9	4	3	2	1	0		
Data validation	A variety of validation/error catching for relevant input Clear and appropriate error messages and exception handling mechanisms	limited validation/error catching for relevant input OR Mostly clear and appropriate error messages and exception handling mechanisms	Limited validation/error catching Error messages and exception handling sometimes inappropriate/ not meaningful	Validation/error catching poorly done or inappropriate/not meaningful	No effort at validation/error catching	4	
Testing □ Test for valid data □ Test for extreme data □ Test for invalid data	Excellent – all three aspects applied correctly in all instances	Good – mostly meaningful for all three aspects with minor shortcomings	Satisfactory – two aspects omitted or not used well	Limited – more than two aspects omitted or not used well	Totally inappropriate or incorrectly applied	4	
						8	

Learner Name:

Task 10	4	3	2	1	0		
Comments/Notes (Explanation of program and code)	Code clearly annotated to fully explain all necessary parts. Explanation shows excellent insight. Extensive project notes present and of an excellent standard. Clearly explains working of the program	Code clearly annotated to explain all necessary parts. Explanation shows good insight. Project notes present and of a very good quality	Code annotated to explain most necessary parts. Explanation shows some insight. Project notes present and of a moderate standard	Code annotated to explain certain parts. Explanation shows little insight Inadequate project notes present	No comments or no project notes	**4**	
Does the program meet the requirements?	Exceeds requirements stated in Phase 1. Comprehensive program. All elements function as specified. Shows insight in all aspects.	Meets the requirements stated in Phase 1. Less comprehensive. All elements function as specified. Shows insight in most aspects.	Meets most of the requirements, but some don't function well Only some program elements function as specified in Phase 1. Shows insight in one or two aspects.	Only meets some requirements, and some don't function well. Basic program. Basic scope. Very limited insight.	Does not meet the requirements. Less than basic. Limited scope.	**4**	
						8	

General: interview

Interview	7	6	4	2	0	
Explain selected code	Explained all selected code clearly and with confidence Shows excellent insight.	Explained selected code with minor shortcomings Shows insight	Unable to explain some of the selected code adequately Shows some insight	Unable to explain most of the selected code, limited insight	Unable to explain any selected code, no insight	
					general	

Assessment Summary

Tasks	Task description	Maximum Mark	Mark Obtained
Task 1	Task definition and user story Acceptance test	15	
Task 2	Database design	12	
Task 3	Navigation / flow between screens GUI design	8	
Task 4	Data dictionary Arrays User defined methods(optional) Text files	12	
Task 5	IPO table and data validation	15	
Task 6	Create GUI application	9	
Task 7	Create DB and connect to Delphi	24	
Task 8	Coding	12	
Task 9	Testing and data validation	8	
Task 10	Documentation	8	
	Subtotal	123	
Interview		7	
	Final mark	**130**	

I hereby declare that the work assessed is solely that of the learner (except where there is clear acknowledgement and record of any substantive advice/assistance given to the learner) concerned and was conducted under supervised/controlled conditions to ensure that the work has not been plagiarised, copied from someone else or previously submitted for assessment by anyone

Comment:

Teacher name: **Teacher signature:** **Date:**

Annexure B

Learner declaration – Task/ ____

I understand that work submitted for assessment must be my own.

Have you received help/information from anyone to produce this work?

☐ No ☐ Yes (provide details below)

Help/information received from (person):	Nature of the help/information (provide evidence):
______________________ Signature of Learner	___ / ___ / 2022 Date

Annexure C

Declaration of authenticity

Learner name		**ID Number**	
Grade	11	**Year**	2022
Subject	**Information Technology**		
Practical Assessment Task (PAT)		**Teacher**	

I hereby declare that the contents of this assessment task are my own original work (except where there is clear acknowledgement and appropriate reference to the work of others) and have not been plagiarised, copied from someone else or previously submitted for assessment by anyone.

__________________________ ___ / ___ / 2022

Signature of Learner Date