



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

INFORMATION TECHNOLOGY

GUIDELINES FOR PRACTICAL ASSESSMENT TASK (PAT)

**Grade 10
2023**

These guidelines consist of 26 pages.

Contents

<i>What is the PAT?</i>	<i>3</i>
<i>Mark allocation</i>	<i>4</i>
<i>The Topic</i>	<i>5</i>
<i>What you need to be able to do the PAT</i>	<i>7</i>
<i>Malpractice</i>	<i>7</i>
<i>Non-compliance</i>	<i>7</i>
<i>PAT requirements</i>	<i>8</i>
<i>Instructions for planning – Task 1 - 5</i>	<i>9</i>
<i>Instructions for implementation – Task 6 - 10</i>	<i>13</i>
<i>Annexure A</i>	<i>16</i>
<i>Assessment Tools</i>	<i>16</i>
<i>Annexure B</i>	<i>25</i>
Learner declaration – Phase ____	<i>25</i>
<i>Annexure C</i>	<i>26</i>
Declaration of authenticity	<i>26</i>

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.
 - Perseverance 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 three stages and **TEN TASKS**:

Task no.	Task description	Stage of system development
Task 0	Problem definition and research (Class discussion)	Requirements
Task 1	Task definition and user story	Requirements
Task 2	Acceptance test	Requirements
Task 3	Navigation / flow between screens	Design
Task 4	Design a screen	Design
Task 5	IPO table and data validation	Design
Task 6	Create TWO Screens in Delphi	Implementation
Task 7	HCI principles for TWO screens	Implementation
Task 8	Develop the Code	Implementation
Task 9	Testing and data validation	Implementation
Task 10	Documentation	Review

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 20% 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	8
Task 2	Acceptance test	4
Task 3	Navigation / flow between screens	4
Task 4	Design a screen	8
Task 5	IPO table and data validation	16
Total Task 1 - 5		40
Task 6	Create TWO Screens in Delphi	4
Task 7	HCI principles for TWO screens	4
Task 8	Develop the Code	20
Task 9	Testing and data validation	8
Task 10	Documentation	4
Total Task 6 - 10		40
General	Final product and impression	20
Final Total:		100

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 PLC, district and provincial level by subject experts.

The Topic

Entrepreneurial Management System/Application

In the current economic environment, it is important for our citizens to think of ways to support themselves. Entrepreneurial initiatives will be necessary to ensure a steady economic growth. Many small businesses are using manual systems to manage their businesses.

For this year, learners are expected to design and develop a Delphi program for a small business that will assist in managing the business.

This project allows you to create a business management system using Delphi. Use good programming principals throughout. Make use of the following pre-scripts:

- Variables of type String / Char, Integer, Real and Boolean
- Name all the components used in the program
- The use of **at least** THREE Forms **OR** a PageControl with **at least** THREE tabsheets (more forms or tabsheets are allowed)
 - Forms / Tabsheets must be used in a way that it will add to the usability of the system.
For example:
 - Form / tabsheet 1 – welcome and login as user/admin
 - Form / tabsheet 2 – buying/taking stock
 - Form / tabsheet 3 – print a report/receipt
- Data validation / Error messages on each of the fields where the user has to key in information
 - EditBox – to test if it is empty; test if it is numbers only (or text only)
 - RadioGroup – test to see if the user has selected an option
 - Buttons – to enable the button only after all the selections were made

This program is targeted to assist a small business user. The focus of the program is to test content knowledge of a subject or to learn new content.

- The management system could be based on an existing system or it could be a new idea.
- The system can be for any small business.
- The complexity of the topics / themes are not awarded more marks in this project.

Examples of different types of business management systems (but not limited to):

- Tuckshop
 - Calculate input costs (price of purchasing stock at cost)
 - Stock take
 - Compiling a receipt, calculate VAT and change
 - Compile a list with unique product code
 - Calculate the rent for the building per square meter
- School clothing store
 - Stock take
 - Compiling a receipt, calculate VAT and change
 - Compile a list with unique product code
 - Print price stickers
 - Order new stock, possible discount option

- Spaza shop
 - Calculate selling price of products
 - Calculate change
 - Calculate the ratio of a recipe if it is double or 3 times
- Delivery service
 - Calculate delivery fee
 - Calculate distance
 - Calculate the time it will take
- Online Shop for home-made items
 - Calculate material costs
 - Calculate how much time it takes to create an item
 - Calculate packaging equipment and marketing (Website/Facebook)
 - Calculate profit
- Barber / Beauty Shop
 - Calculate costs of products used per person, according to what they want
 - Calculate running costs (Water, rent, employee-fees)
 - Calculate profit
- Repair service (Car, computer, etc.)
 - Calculate costs of products used per person, according to what they required
 - Calculate running costs (Water, rent, employee-fees)
 - Calculate profit
- Pet/House sitting Service
 - Calculate hourly rate for which type of pet / house
 - What will you need to make sure the job is well done.
 - Marketing costs must be calculated
 - Which pets needs what attention, type of food, etc.

Your final program must comprise of **ONE** single, logically related piece of software. Projects that consist of two or more unrelated programs will only obtain marks for **ONE** of the parts since only **ONE** of the programs will be regarded as the actual project.

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 software.
- Storage media to save and backup your work electronically, for example a flash drive or online / cloud storage.

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 acknowledging it.

The above actions constitute malpractice, for which a penalty will be applied, depending on the severity 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 for IT. This will result in an incompleteness of your promotion mark, and it may result in you not passing your grade.

PAT requirements

The project must include the following:

- A GUI with good functionality and usability, based on sound HCI principles.
- Other data structures that will be relevant to your program.

GUI

The graphical user interface (GUI) must

- have at least **THREE** tabsheets that allows for navigation between forms depending on the user choices.
- comply with relevant HCI principles.

OPTIONAL DATA STRUCTURES: OUTSIDE GRADE 10 CURRICULUM:

- **Text files**

Your application could use a text file(s) for input and / or output using text stream operations. (LoadFromFile, SaveToFile)

- **Arrays**

Your application could use an array to keep track of input and / or output data, for example: stock, personal, suppliers.

- **Dynamic objects**

Dynamic objects can be used in your application to create your own components.

Instructions for planning – Task 1 - 5

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	What type of entrepreneurial management system was selected by you, the programmer	
Purpose of program	Describe the purpose of your program – why does the small business owner need your program. Research a few different small business operations and write down the positives of each small business.	
Possible solution	What will the program do to meet the identified needs? Describe how your program will work. Include a description of each of your tab sheets and how the user will interact with your program. Include the positives of best practices into your business management system.	
Scope	Explain what limitations your program may have.	
	Proposed total time	60 – 90 min

Task 1a Define the Task

Write a brief description (approximately 150 words) in your own words to describe, in general terms, the problem/task and how the project will solve the problem.

Use the template provided: see **Annexure 1**.

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

The **user** is the target audience and will thus dictate the needs and requirements of the program. In this task we will determine who the users / owners are and their specific requirements.

The aim is to identify the user(s), user needs and processing requirements of the system. Use a template for the user story board to explain the role, activity of each user of the system. (Use the template provided: see **Annexure 1**.)

Note:

*The **user** is the target audience, user of the program, owner of the small business, etc.*

*The **user story** is told by the user and formulates in a sentence or two, using everyday language, what he / she wants to be able to do with the program.*

For example:

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

<i>WHO</i>	<i>WHAT</i>	<i>WHY</i>
<i>As a ...</i>	<i>I want to ...</i>	<i>So that ...</i>
<i>User/Actor/role</i>	<i>Goal/program feature required</i>	<i>Value or benefit</i>

Verb and noun analysis:**Nouns:**

- Learner: The person who wants to register and play the Maths game.
- Register: The action the learner wants to take to create an account or profile.
- Maths game: The game the learner wants to play to improve their skills.

Verbs:

- Want: Indicates the learner's desire to complete a certain action.
- Register: The action the learner wants to take in order to create an account or profile.
- Play: The action the learner wants to take in order to engage in the Maths game.
- Improve: The outcome the learner hopes to achieve by playing the Maths game - that their Maths skills will become better.

Task 2 Acceptance test

An acceptance test – use the user stories to identify the goals that represent a functionality (functional requirement) that can be used or performed in isolation. For example, how does the programmer know that the user stories have been satisfied.

“The system shall ...”(Use the template provided: see **Annexure 2.**)

- e.g. The system shall ask the user for a number
 The system shall randomly select a question
 The system shall display the answer

Note: A total of FOUR acceptance tests need to be identified from the TWO user stories.

Task 3 Navigation/ Description of Flow Diagram

Clearly indicate the logical program flow and navigation between screens. Use the provided template to illustrate a global overview of the project/system. More screens to be added on the template and the flow clearly indicated, between the different screens. (Use the template provided: see **Annexure 3.**)

Task 4a Screen Design

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.

Provide examples of planned data capture and data entry designs. Use the provided templates to present **TWO screen** designs and the planned output design **OR TWO screen** dumps in Delphi may be used as a prototype screen. (See **Annexure 4**).

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

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 for example: Animation / Graphics / Components.

Task 4b Data Dictionary

A data dictionary is used to describe where the programmer will use different variables, components, and data structures.

VARIABLES AND COMPONENTS:

Your application should contain a variety and correct use of appropriate:

- variables types and components (at least **THREE** data types)
- local and global variables
- naming conventions for variables and components

OPTIONAL DATA STRUCTURE OUTSIDE GRADE 10 CURRICULUM

- **Text files**
Your application could make use of 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**
Your application could use an array to keep track of input and / or output of data. Explain the purpose and implementation of the array and how it will add value to the program.
- **Dynamic components**
Your application could make use dynamic components. Explain how you would make use of dynamic components in your program so that it adds value to the program.

Task 5 Input, Processing, Output (IPO) table and data validation

Use an IPO illustration/table provided (See **Annexure 5**) to:

- Specify the:
 - format
 - data types
 - source of input
 - source of output,
 - validation of input
 - error checking mechanisms.
- Specify processing that needs to be done, provide algorithm(s) / formulae to show how the processing will be done (minimum of **TWO** algorithms).
- Provide a clear description to indicate the input.
- Provide the output requirements of the system for at least **TWO** of the main interfaces.

Summary: Hand-in documents – Task 1 to 5

Hand in a document that contains the following, compiled according to the detailed information provided in this:

- **Annexure 1:** Completed template with a clear description of the chosen topic. The user requirements - detailed information stating the role, activities of the system (Task 1a and task 1b)
- **Annexure 2:** Limitations of each user of the planned system **OR** a list of system activities beginning with the words: “The system shall ...” (Task 2)
- **Annexure 3:** Navigation of screens on the completed template (Task 3)
- **Annexure 4:** **TWO** screen designs (GUI Design) completed template **OR TWO** Delphi screen shots. (Task 4a)
- **Annexure 5:** The IPO design including validation template (Task 5)

Instructions for implementation – Task 6 - 10

This is where you implement your design by using appropriate software tools (programming language, IDE, etcetera) and techniques to construct a solution to the problem.

Task 6 Create TWO Screens in Delphi

Developing the GUI according to the planning documents that was developed during the requirement stage. Use appropriate components to ensure easy use and effective navigation.

Task 7 HCI principles for TWO screens

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.

Screens 1 and 2 are different screens to be used in the coding solution.

Task 8 Develop the Code

Write code to develop the program / system according to the planning documents 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 loops and conditional structures.
- **Optional** datastructure outside the Grade 10 curriculum:
 - Text file
 - Array
 - Dynamic component
- Input data using the most effective method, for example a text file, keyboard, components
- Process the data making use of the appropriate methods.
- Generate output of data using the correct components and structures, include formatting where needed.
- Ensure smooth interaction between forms / tabs.

Task 9 Test and validate the program

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

Validate if the program is working accurately using the test data

1. Isaacs (2021) discusses essential software testing test every software engineer must do before showcasing their work to others: Basic factuality testing

Goal: Do not let anybody touch your work if it is going to crash as soon as they enter their own name into the username field.

- Use relevant validation procedures and components.
- Check if every button on the screen works.
- Ensure that you can enter simple text into each field without crashing the software.
- Try different combinations of clicks and characters.

2. Code review

It is good practise to let another pair person check the source code to uncover possible mistakes.

Task 10 Documentation

Documentation is undoubtedly the activity that we all like to do the least. Unfortunately, writing documentation is not fun: it is a need. Due to inadequate documentation up to 60% time is spent to understand the existing program. Good documentation helps people to understand and integrate into the project much faster

Write project notes:

- Describe what the program does
- Describe how to use / interact with the program

Summary: Hand-in documents – Task 6 to 10

Hand in:

- The completed Delphi project (Delphi code, text files (if used - optional) and any other resources required to execute the program successfully).
- The documentation of the program (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

Phase 1		Learner name:						
Task 1A: Define the Task	4	3	2	1	0		Mark	Mod
Topic / task definition: <ul style="list-style-type: none"> • Topic is clearly stated • Thorough description of what the problem/task involves (purpose) • Describe a possible solution for the problem/task • A description of the scope of the project is provided. 	An excellent presentation of all FOUR points listed	All FOUR points were presented with shortcomings OR A good presentation of THREE points	THREE points were presented with shortcomings OR A good presentation of TWO points	TWO points were presented with shortcomings OR A good attempt to present ONE of the points	Totally inadequate or not applicable Poor or no coverage of the aspects No scope or extremely vague and unclear	4		
Task 1B: User Story	4	3	2	1	0			
(Use Case diagram OR table format) Role, activity, value (who, what, why) <ul style="list-style-type: none"> • 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		
Task 2: Acceptance test	4	3	2	1	0			
Acceptance test for the users: To complete the sentence with at least FOUR instructions: "the system shall ... "	An excellent presentation of all FOUR instructions	All FOUR instructions were presented with shortcomings OR A good presentation of THREE instructions	THREE instructions were presented with shortcomings OR A good presentation of TWO instructions	TWO instructions were presented with shortcomings OR A good attempt to present ONE of the instructions	Totally inadequate or not applicable Poor or no coverage of the any instructions	4		

Phase 1**Learner name:**

Task 3: Navigation between screens	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		
Task 4A: Screen Design	4	3	2	1	0			
Planning tab sheet 1 + 2 Design fits to program's intended use • Appropriate components • Clearly marked navigation • Friendly dialogue / Help	All FOUR principles applied appropriately	All FOUR principles applied with minor errors	THREE principles applied appropriately	TWO principles applied appropriately	GUI design not functional OR Does not support the intended use at all	4		
Task 4B: Data Dictionary	4	3	2	1	0			
Variables and Components: • Correct planning for the variety / appropriate components • Correct planning for the variety / appropriate variable types • Correct planning for the use of local and global variables • Proper naming convention of variables and components	Excellent – All four aspects applied correctly	Good – Three aspects applied correctly	Satisfactory – Two aspects applied correctly	Limited – One aspect applied correctly	Totally inappropriate or incorrectly applied	4		

Task 5: IPO and Validation	4	3	2	1	0			
Input interfaces (at least TWO interfaces) <ul style="list-style-type: none"> • Data type (At least THREE overall) • Format of the input, e.g. date, gender (M/F) • Most suitable input component 	Clearly describes all inputs according to all THREE points listed for THREE data types	Minor shortcomings in describing all inputs according to all THREE points listed for TWO data types	Major shortcomings in describing all inputs according to THREE data types	Poor attempt to describe input values	No inputs described OR Incorrect	4		
Input validation <ul style="list-style-type: none"> • At least THREE different data types validated • At least THREE inputs validated including: <ul style="list-style-type: none"> ž Validate for NULL/empty field OR Test if value was selected in a selection component • Associated error messages 	Clearly describes all points listed	Clearly describes THREE points listed OR Minor shortcomings in describing all points listed	Clearly describes TWO points listed OR Major shortcomings in describing all points listed	Poor attempt to describe validation	No validation described OR Incorrect	4		
Data processing/ Algorithms WHAT processing will need to be done (provide TWO) HOW processing will be done – supply algorithms, formulas, etc. (Describe TWO algorithms)	ž List at least TWO processes to be done ž Clearly describes at least TWO algorithms correctly	One process listed 'Clearly describes ONE processes correctly	A poor attempt to describe how TWO algorithms/processes will be done.	A poor attempt to describe one processes/algorithms	No processes listed Processes not described/incorrect or irrelevant	4		
Data output interfaces (at least TWO) • Data to output• Format of the output, e.g. currency, date• Most suitable Output component, such as panel, rich edit, label, etc.ž Minimum TWO output components	Clearly describes all outputs by addressing all FOUR points listed	Minor shortcomings in describing all outputs by addressing THREE points listed	Clearly describes all outputs by addressing TWO points listedOR Limited outputs described	Poor attempt to describe outputs	No output describedOR Incorrect	4		
Total						40		

Phase 2**Learner name:**

Task 6 - Both Screens - coding	4	3	2	1	0		Mark	Mod
*Variety of components *Navigation on screen (flow of events) *Feedback / Help *Labelling	Excellent – all four aspects applied correctly throughout the program	Good – one aspect omitted or not applied well	Satisfactory – two aspects omitted or not applied well	Limited – more than two aspects omitted or not applied well	Poor GUI design Little/No thought given to HCI principles	4		
Task 7 - HCI principles for both screens	4	3	2	1	0			
* Consistency * Navigation between screens * Informative feedback * Easy reversal of action i.e. Reset	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		
Task 8 - Coding	4	3	2	1	0			
Variables • Variety of appropriate variable types • Correct use of local and global variables • Proper naming convention of variables, e.g. iNumber, sName • Correct prefix for components, e.g. edt, red, cmb	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		
Input data • Variety of sources of input, such as from the keyboard, mouse click and computer generated • Correct data types • Appropriate format used, e.g. date, gender (M/F) • GUI component used	Excellent application of all FOUR aspects listed	Minor shortcomings in the application of all THREE aspects listed	Approximately 50% of the aspects listed correctly applied	Limited application of the aspects listed	No application of the aspects listed	4		

Phase 2**Learner name:**

Task 8 - Coding	4	3	2	1	0			
Process Use the following: * Built-in Function * Nested-If statement * IN operator * User-defined function	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		
Algorithm correctness/Processing • Correct implementation of Input and Output accordingly • Correct implementation of using Processing accordingly • Outside curriculum data structures correctly applied • Excellent use of outside curriculum data structures	All algorithms used are appropriate, work correctly and meet all processing requirements	Appropriate algorithms that work correctly but ONE processing requirement not met	50% of the algorithms used are appropriate, work correctly and meets most processing requirements	Algorithms are mostly inadequate/mostly not working correctly, processing requirements not all met	Totally inadequate OR Not working correctly	4		
Output • Layout • Readability/Clarity, eg. columns, headings • Formatted, eg. currency • Most appropriate component/ data structure used for output	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		
Task 9 - Verification and Testing	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		

Phase 2**Learner name:**

Task 9 - Verification and Testing	4	3	2	1	0			
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		
Task 10 - Documentation	4	3	2	1	0			
Comments/Notes (Explanation of program and code) • Good programming techniques • Comments to explain all necessary parts	All Variable, constant and component names well chosen to make for readable codeComments explain all the necessary parts	All Variable, constant and component names well chosen to make for readable code, some shortcomingsComments explain all the necessary parts, some shortcomings	Some variable, constant and component names well chosen AND /ORComments explain some of the necessary parts	Less than 50% of variable, constant and component names well chosen AND /ORLess than 50% of comments explain some the necessary parts	No comments or no project notesCode not readableAnd no comments	4		
Total						40		

General: Final product and impression.**Learner name**

Aspect	4	3	2	1	0		Mark	Mod
Completeness	Reached initial goal and met all stated requirements in Phase 1	Met at least 80% of the initial requirements	Met more than 50% of requirements	More than 50% of initial requirements not met	Almost none of the initial requirements met	4		
Professional product	Useful and can be implemented as a real-life application Well-designed and user-friendly Contains no errors	Useful as real-life application with minor adjustments Good design and user-friendly Contains minimal errors	Useful as real-life application with major adjustments Good design and user-friendly Contains several errors	Not ready to be implemented as real-life application, but has some potential	Not ready to be implemented as real-life application Poor design	4		
Attitude and commitment	Kept to due dates. Well-designed phases. Showed exceptional commitment and pride in work done.	Kept to due dates. Phases designed at an acceptable level. Showed commitment and pride in work done.	Kept to the due date for one of the phases. One of the phases not developed at an acceptable level. Show some commitment.	Both phases not handed in on time/poorly designed. Displayed a lack of commitment.	Phase 1 and Phase 2 was not handed in. Showed no commitment.	4		
Interview	8	6	4	2	0			
Ability to explain code	Explained all selected code clearly and with confidence Shows excellent insight.	Explained the selected code with minor shortcomings Shows some 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	8		
Total Mark						20		

Learner name:**Assessment Summary**

Phase	Task	Focus	Maximum Mark	Mark Obtained	Mark Moderated
1	Task 1	Task definition and user story	8		
	Task 2	Acceptance test	4		
	Task 3	Navigation / flow between screens	4		
	Task 4	Design a screen	8		
	Task 5	IPO table and data validation	16		
2	Task 6	Create TWO Screens in Delphi	4		
	Task 7	HCI principles for TWO screens	4		
	Task 8	Develop the Code	20		
	Task 9	Testing and data validation	8		
	Task 10	Documentation	4		
3	General	Finale product and impression	20		
Total			100		
Final mark (100%)					
Final mark (25%)					

Declaration of Authenticity – FINAL PAT

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/feedback:

Teacher name: _____ Teacher signature: _____ Date: _____

HOD signature: _____ Date: _____

Teacher Moderator signature: _____ Date: _____

DSA signature: _____ Date: _____

Annexure B

Learner declaration – Phase _____

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

___ / ___ / 2023

Date

Annexure C

Declaration of authenticity

Learner name		ID Number	
Grade	10	Year	2023
Subject	Information Technology		
Practical Assessment Task (PAT)		Teacher	
<p>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.</p>			
<p>_____</p> <p>Signature of Learner</p>		<p>___ / ___ / 2023</p> <p>Date</p>	