basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

# INFORMATION TECHNOLOGY

## GUIDELINES FOR

## PRACTICAL ASSESSMENT TASK (PAT)

## Grade 11

## 2024

**These guidelines consist of 32 pages.**

# Contents

*Contents* 2
*What is the PAT?* 3
*Mark allocation* 5
*Topic* 6
*What you need to be able to do the PAT* 8
*Malpractice* 8
*Non-compliance* 8
*PAT requirements* 9
*Instructions* 10
*Annexure A: Assessment tools* 18
*Annexure B* 31
Learner declaration – Task/ ____ 31
*Annexure C* 32
Declaration of authenticity 32

# 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 **TEN TASKS** as named below:

| **Phases** | **Task no.** | **Task description** | **Stage of application development** |
|---|---|---|---|
| | Task 0 | Problem definition and research | Requirements |
| **Phase 1** | Task 1 | Task definition, user story and acceptance test | Requirements |
| | Task 2 | Data dictionary<br>• Arrays<br>• Text files<br>User-defined methods | Requirements |
| | Task 3 | Data dictionary<br>• Database design | Requirements |
| | Task 4 | Navigation / flow between screens and GUI design | Requirements |
| | Task 5 | IPO table and data validation | Design |
| **Phase 2** | Task 6 | Create GUI application | Design |
| | Task 7 | Create a Database and connect to application | Implementation |
| | Task 8 | Coding | Implementation |
| | Task 9 | Testing and data validation | Implementation |
| **Phase 3** | Task 10 | Documentation and Interview | Review and 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 various interview sessions as you progress.**

# 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.

| Phases | Tasks | Task description | Maximum Mark | Grouping |
|---|---|---|---|---|
| Phase 1 | Task 1 | Task definition, user story and acceptance test | 12 | |
| | Task 2 | Data dictionary<br>• Arrays<br>• Text files<br>User defined methods | 16 | |
| | Task 3 | Data dictionary<br>• Database design | 12 | |
| | Task 4 | Navigation / flow between screens and GUI design | 4 | |
| | Task 5 | IPO table and data validation | 20 | 64 |
| Phase 2 | Task 6 | Create GUI application | 6 | |
| | Task 7 | Create a Database and connect to application | 6 | |
| | Task 8 | Coding | 50 | |
| | Task 9 | Testing and data validation | 8 | 70 |
| Phase 3 | Task 10 | Documentation and Interview | 16 | 16 |
| | | **Total:** | **150** | |

**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

## HEALTHY LIVING PROGRAM

Healthy living means making choices that result in a healthy body and mind. A person can maintain a healthy lifestyle by eating a healthy diet and being physically active. For the body to function properly, it needs nutrients. For this year, learners are expected to design and develop a Delphi program for an interactive "**Healthy Living Program**".

Projects in this scenario could include the following topics related to **healthy living**:

- Exercise program
- Diet program
- Improving sales of healthier products
- Provide a program to people living with medical conditions
- Develop an obesity prevention program
- Healthy living loyalty program
- Personal health data manager
- Recipes for healthy dishes

Choose an application related to **ways for healthy living** and do research on the information system requirements.

You are not limited to the list of ideas above, but you need to keep within the overall theme of **healthy living. Note** that you need to choose data and functionalities (services) in such a way as to develop a well-rounded application related to the topic.

This **healthy living** software program is targeted at someone who would like to improve or support a healthy lifestyle.

**NOTE**: Your final program must comprise one single program with logically related parts.

**Resources:**

Here is a list of some resources that can be used to determine your PAT topic:

- MyPlate – free tools and resources to meet your unique needs
  - MyPlate
- BetterHealth Channel – provides health and medical information to improve the health and wellbeing of people and communities they live in.
  - BetterHealth
- Eat Better South Africa NPC is dedicated to supporting our local public healthcare systems and fostering sustainable food systems.
  - Eat Better SA
- A Cash-Back Rebate Program for Healthy Food Purchases in South Africa
  - National Institute of Health
- The National Heart, Lung, and Blood Institute
  - Health Education
- Cooking from the heart
  - My dynamics

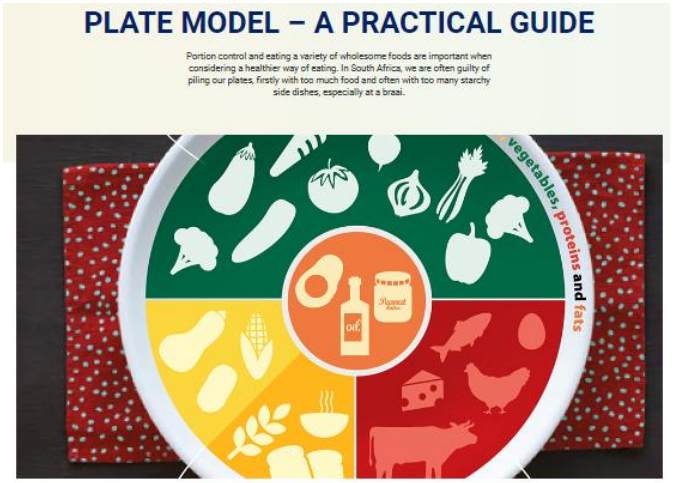
- The Heart and Stroke Foundation
  - Healthy Living > Healthy weight
- South African-Based Childhood Obesity Prevention Programme
  - Obesity
- Healthy children – powered by paediatricians, trusted by parents
  - Are your children getting enough exercise each day?

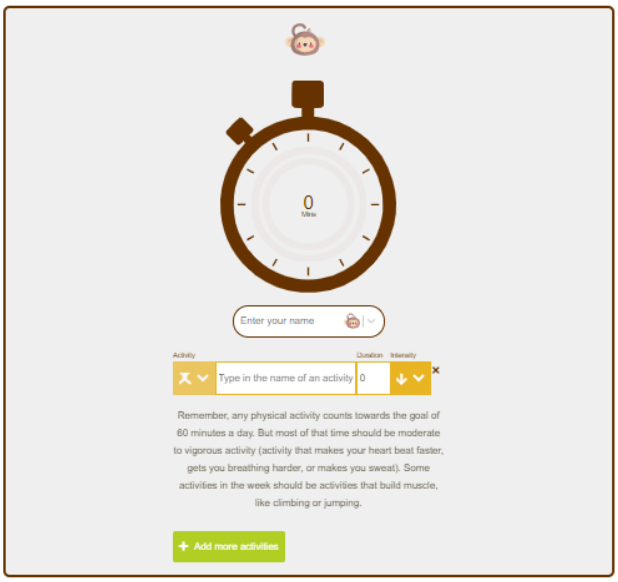
# Healthy living examples:



How do you know if your weight is too high?

| | BMI (kg/m²) | Waist circumference | Approx. pants size |
|---|---|---|---|
| Underweight | < 18.5 | | |
| Normal | 18.5 - 24.9 | Men ≤ 94 cm / Women ≤ 80 cm | Men ≤ 36 / Women ≤ 40 (US 14) |
| Overweight | 25.0 - 29.9 | Men > 94 - 102 cm / Women > 80 - 88 cm | Men ≥ 38 / Women ≥ 42 (US 16) |
| Obesity | > 30.0 | Men > 102 cm / Women > 88 cm | Men ≥ 42 / Women ≥ 44 (US 18) |



# 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, 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 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 e.g. Arrays and user-defined methods 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 structures to provide the necessary input, processing and output
- comply with relevant HCI principles

**Text files**

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

**Arrays**

The appropriate use of array(s).

**User defined methods**

Your application needs to make use of user defined methods.

**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 program 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 Healthy Living Program.
- **Purpose of program**
  Describe the purpose of your program – why does the organiser need your program.
  Research a few different Healthy Living programs and write down the positives of each.
- **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 can do AND what your programs limitations are.

## INTRODUCTION: TASK 1 - 5

During the following tasks you need 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 program 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

**User stories:**

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 program. Use a table or a 'use case diagram' to explain the role and activity of each user of the program.

For example:

> **As a** learner **I want to** register on a fitness program **so that** I can improve my health and wellbeing.

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

Verb and noun analysis example:

Nouns:

- Learner: The person who wants to register on a fitness program to improve their health and wellbeing.
- Register: The action the learner wants to take to create an account or profile.
- Fitness program: A program that allows a person to improve their health and wellbeing.

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.
- Improve my fitness and wellbeing: The outcome the learner hopes to achieve by participating in the fitness program.

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.

From the example user story listed above, when does the programmer know the following:

- A learner registered.
- A learner can see their fitness and wellbeing programs progress.
- A learner can improve their fitness.

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

## TASK 2: 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:

- variable types and components
- local and global variables
- naming conventions for variables and components

**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.

**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.

**User defined methods**

Your application must make use of user defined methods. Explain where you will create methods that will be reused in your program to add value to the program.

**Other data structures**

Your application may include other data structures such as parallel/two-dimensional arrays, classes, and objects, and applying programming concepts such as polymorphism, overloaded methods, method binding, etc.

**Note:** Your application must make use of a database. Task 3 will focus on the database aspects.

## TASK 3: Database design

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. A description of the database and how data will be manipulated in the program should be included.

Show the design of the database, including at least TWO tables with field names, field types, field sizes and a description of the fields.

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, etc.

**NOTE:** A relational database is optional, and the relationship between the tables should then clearly be indicated.

## TASK 4: Navigation / flow between screens
## GUI design

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.

## TASK 5: Input, Processing, Output (IPO) table 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.
- Specify the following aspects of input (at least **TWO** interfaces):
    - Format data types, source and component used.
    - Validation of input:
        - At least **FOUR** different input validation data types:
        - At least **FOUR** input validation types:
            - Validate for NULL/empty field
            - Test if value was selected in a selection component.
    - Error checking mechanisms – data types and inputs validated with associated messages.
- Specify the layout, readability, format, data types, and appropriate output.
- Specify processing that needs to be done of at least **EIGHT** processes and provide algorithm(s)/formulae to show how the processing will be done of at least **FOUR** processes.
- Provide a clear description to indicate the input, processing and output requirements of the program 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 (screenshots) and of planned output design.

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

**NOTE:** Screenshots may be used from a prototype of the project but must be annotated.

## TASK 7: Create a Database and connect to application

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** *tblRecipies*.
- 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: Coding

Write code to develop the program 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 / methods are compulsory and in addition to the database:
    - Text file - reading OR writing OR appending
    - Array(s)
    - User defined methods
- The use of any other data structures not already tested / advanced programming construct / dynamic component.
- Use relevant validation procedures and components.
- Develop a well-designed and user-friendly GUI.
- Rename relevant components to add to readability of your code.
- Input data using the most effective method, for example a text file, database, keyboard, components etc.
- 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 the data in the database (CRUD).

## TASK 9: Testing and data validation

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

Marks will be allocated for the validating of coding 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<br>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: Documentation and Interview

**Documentation:**

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 added as comments. 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.

**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.

## 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).**

# Annexure A: Assessment tools

| **Learner Name:** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task 1A** | **4** | **3** | **2** | **1** | **0** | | |
| **Define the Task**<br>(Short description ±150 - 200 words)<br>• Task is clearly described<br>• The need of the task is identified.<br>• The possible solution will solve the task.<br>• A description of the scope of the project is provided. | Excellent – All FOUR aspects covered | Good – THREE aspects covered | Satisfactory – TWO aspects covered | Limited – Only ONE aspect covered | No aspects covered | **4** | |
| **Task 1B** | **4** | **3** | **2** | **1** | **0** | | |
| User story: (Use Case diagram **OR** table format)<br>Role, activity, value<br>(who, what, why)<br>• Who will use the system?<br>• What are the goals/ activities that user will perform?<br>• Why do they want/need it? | Role, activity, value of all users (at least TWO different types of users) of the system thoroughly and correctly described.<br>Well documented, clear and to the point. | Role, activity, value of all users (at least TWO different types of users) of the system described but minor shortcomings e.g. one instance where goal is not clear, value not clear, etc.<br>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.<br>Only ONE type of user of the system discussed.<br>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<br>Poorly documented – not acceptable | Not done or incorrect or irrelevant | **4** | |
| User stories (Acceptance Tests)<br>List of functions the program should perform | Detailed list which clearly and correctly defines at least FIVE functions<br>All functions derived from the user stories | Fairly detailed list which clearly and correctly defines FOUR functions<br>At least THREE functions derived from the user stories | A list which defines THREE functions containing some detail<br>Only TWO functions derived from user stories | A list which defines TWO functions but lacking in detail<br>Only ONE function derived from user stories | No list of functions included<br>Very rudimentary, no detail<br>Not derived from user stories | **4** | |
| | | | | | | **12** | |
| **Comments and feedback:** | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| **Learner Name:** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task 2: Data Dictionary** | **4** | **3** | **2** | **1** | **0** | | |
| **Variables and components**<br>• Correct planning for the variety / appropriate components<br>• Correct planning for the variety / appropriate variable types<br>• Correct planning for the use of local and global variables<br>• 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** | |
| **Data Structures (Excl. Database)** | **4** | **3** | **2** | **1** | **0** | | |
| **Text files(s)** | Excellent and relevant planning for the use of a text file | Good planning for the use of a text file | Limited planning for the use of a text file. | An attempt to plan for the use of a text file with short comings. | Not done or irrelevant. | **4** | |
| **Array(s)** | Excellent and relevant planning for the use of an array | Good planning for the use of an array | Limited planning for the use of an array with minor shortcomings | An attempt to plan for the use of an array with many short comings. | Not done or irrelevant. | **4** | |
| **User defined methods** | Excellent and relevant planning for the use of User Defined Methods | Good planning for the use of User Defined Methods | Limited planning for the use of User Defined Methods with minor shortcomings | An attempt to plan for the use of User Defined Methods. Shows potential but is not planned correctly. | Not done or irrelevant | **4** | |
| | | | | | | **16** | |
| **Comments and feedback:** | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| **Learner Name:** | | | | | | |
|---|---|---|---|---|---|---|
| **Task 3: Data Dictionary - Database** | **3** | **2** | **1** | **0** | | |
| **Role of DB**<br>(***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.)<br>Role of manipulation in program described/motivated | Manipulation and interaction well described and well-motivated.<br>Most appropriate to meet requirements | Manipulation and interaction not clearly described or substantiated in one instance.<br>Adequate to meet requirements | Manipulation and interaction not well described or motivated in two instances.<br>Mostly not meeting requirements | No database or incorrect or irrelevant or no interaction | **3** | |
| **Choice of fields** | Well-chosen fields. All fields contribute to the solution<br>**AND**<br>Contains no field that could be calculated from other data | One field does not contribute to the solution<br>**OR**<br>One field that could be calculated | More than one field do not contribute to the solution<br>**OR**<br>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**<br>• At least 2 tables<br>• Tables have fields that are correctly related.<br>• Primary key correctly assigned | Excellent – All THREE aspects clearly covered | Good – TWO aspects clearly covered | Satisfactory – Only ONE aspect clearly covered | No database. | **3** | |
| | | | | | **12** | |
| **Comments and feedback:** | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| **Learner name** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task 4: Navigation / flow between screens & GUI design** | | | | | | | |
| | **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<br>**OR**<br>Incorrect, irrelevant or unsuitable for the application | **4** | |
| | | | | | | **4** | |
| **Comments and feedback:** | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| **Learner Name:** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task 5: IPO – Software design tool** | | | | | | | |
| | **4** | **3** | **2** | **1** | **0** | | |
| **INPUT: Interfaces (at least TWO)**<br>• Source of input, such as from the keyboard, text file, array or database<br>• Data type<br>• Format of input, e.g. date, gender (M/F)<br>• GUI component used | Clearly describes all inputs according to all FOUR points listed | Minor shortcomings in describing all inputs according to all FOUR points listed | Clear description according to THREE points listed<br>**OR**<br>Major shortcomings in describing all inputs according to all FOUR points listed | Poor attempt to describe input values | No inputs described<br>**OR**<br>Incorrectly described | **4** | |
| **INPUT: Validation**<br>• At least FOUR different data types validated<br>• At least FOUR inputs validated<br>• Associated error messages | Clearly describes all points listed | Clearly describes TWO points listed<br>**OR**<br>Minor shortcomings in describing all points listed | Clearly describes ONE point listed<br>**OR**<br>Major shortcomings in describing all points listed | Poor attempt to describe validation | No validation described<br>**OR**<br>Incorrectly described | **4** | |
| **PROCESSING**<br>What processing will need to be done | List at least EIGHT processes | List SIX to SEVEN processes | List THREE to FIVE processes | Only ONE or TWO processes listed | No processes listed | **4** | |

| **Learner Name:** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task 5: IPO – Software design tool** | | | | | | | |
| | **4** | **3** | **2** | **1** | **0** | | |
| **PROCESSING**<br>How processing will be done – supply algorithms, formulas, etc. | Clearly describes how at least FOUR processes will be done | Clearly describes how THREE processes will be done | Clearly describes how TWO processes will be done **OR** An attempt to describe how FOUR processes will be done | Clearly describes how ONE process will be done **OR** A poor attempt to describe TWO or THREE processes | Processes not described or incorrect or irrelevant | **4** | |
| **OUTPUT: Interfaces (at least TWO)**<br>• Data to output<br>• Format of the output, e.g. currency, date<br>• Output component, such as dbGrid, rich edit, label, etc. | Clearly describes all outputs by addressing all THREE points listed | Minor shortcomings in describing all outputs by addressing all THREE points listed | Clear description of all outputs by addressing TWO points listed<br>**OR**<br>Limited outputs described | Poor attempt to describe outputs | No output described<br>**OR**<br>Incorrectly described | **4** | |
| | | | | | | **20** | |
| **Comments and feedback:** | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| **Learner Name:** | | | | | | |
|---|---|---|---|---|---|---|
| **Task 6** | **3** | **2** | **1** | **0** | | |
| **HCI principles**<br>(***How*** GUI will meet the users' needs). Does it consider:<br>• Purpose of program and user<br>• Standard GUI design principles<br>• Ease of use, logic flow<br>• Clearly marked navigation<br>• Friendly dialogue<br>• Helpful error messages | Good GUI design, considering almost all (at least FIVE) of the principles for at least TWO of the main interfaces, excluding the introductory screen. | Satisfactory GUI design, considering most (at least FOUR) of the principles for at least TWO of the main interfaces, excluding the introductory screens. | Limited GUI design, considering only 50% (at least THREE) of the principles for at least two of the main interfaces, excluding the introductory screens. | Poor GUI design considering less than 50% (less than TWO) of the principles considered. | **3** | |
| **Components** | Appropriate components used in all cases | In ONE or TWO cases another component would have been more appropriate | In THREE or FOUR cases another component would have been more appropriate | Inappropriate components used in more than FOUR cases. | **3** | |
| | | | | | **6** | |
| **Comments and feedback:** | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| **Learner Name:** | | | | | | |
|---|---|---|---|---|---|---|
| **Task 7** | **3** | **2** | **1** | **0** | | |
| **Construct the database**<br>• Appropriate naming convention of tables, fields.<br>• Appropriate field sizes<br>• Appropriate data types for fields | Excellent – All THREE aspects clearly covered | Good – TWO aspects clearly covered | Satisfactory – Only ONE aspect clearly covered | No database. | **3** | |
| **Connecting the database**<br>• Connect DB with ADO/similar connection/code<br>• Connect tables with appropriate component/code<br>• Connect data source with appropriate component/code | Excellent – All THREE aspects clearly covered | Good – TWO aspects clearly covered | Satisfactory – Only ONE aspect clearly covered | No database. | **3** | |
| | | | | | **6** | |
| **Comments and feedback:** | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| **Learner name:** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task 8** | **4** | **3** | **2** | **1** | **0** | | |
| **Variables and components**<br>• Variety of appropriate variable types<br>• Correct use of local and global variables<br>• Proper naming convention of variables, e.g., iNumber, sName<br>• 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** | |
| **Text files(s)** | Effective, excellent and relevant use of a text file(s) | Good use of a text file(s) | Limited use of a text file | An attempt to use a text file with shortcomings | Not done | **4** | |
| **Arrays** | Effective, excellent and relevant use of arrays OR parallel arrays | Good use of array(s) OR parallel arrays | Limited use of array(s) with minor shortcomings | An attempt to use an array. Shows potential but not used for a suitable purpose or does not work correctly. | Not done | **4** | |
| **User defined methods** | Effective, excellent and relevant use of User Defined Methods | Good use of User Defined Methods | Limited use of User Defined Methods with minor shortcomings | An attempt to use User Defined Methods. Shows potential but not used for a suitable purpose or does not work correctly | Not done | **4** | |
| **Input data**<br>• Variety of sources of input, e.g. from the keyboard, text file, array or the database<br>• Correct data types<br>• Appropriate format used, e.g. date, gender (M/F)<br>• Correct GUI component used | Excellent application of all FOUR aspects listed | Minor shortcomings in the application of all FOUR aspects listed | Approximately 50% of the aspects listed correctly applied | Limited application of the aspects listed | No application of the aspects listed | **4** | |
| **PROCESSING**<br>Algorithm correctness/ Processing | 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/not working correctly, processing requirements not all met | Totally inadequate or not working correctly | **4** | |

| **Learner name:** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task 8** | **4** | **3** | **2** | **1** | **0** | | |
| **PROCESSING**<br>Algorithm efficiency | All algorithms provide the most efficient solution.<br>Good programming techniques used.<br>Effective modular design with correct use of own functions and procedures | Most algorithms provide the most efficient solution.<br>Acceptable programming techniques used.<br>Limited modular design with correct use of own functions and procedures. | Limited efficiency of algorithms used.<br>Few algorithms use good programming techniques.<br>Poor modularity with limited use of own functions and procedures. | Poor efficiency of algorithms used.<br>Algorithms do not use good programming techniques.<br>Attempted use of own functions and procedures. | Totally inadequate or not working correctly | **4** | |
| **PROCESSING**<br>Relevant and appropriate use of complex code, e.g. Dynamic component | Excellent use of complex code that works correctly.<br>Adds value to the system | Works correctly.<br>Adds value to the system | Works correctly with minor shortcomings. | An attempt has been made with major shortcomings. | No attempt has been made. | **4** | |
| **OUTPUT**<br>• Layout<br>• Readability/Clarity, e.g., columns, headings<br>• Formatted, e.g., currency<br>• Most appropriate component/ data structure used for output | Excellent application of all FOUR aspects listed | Minor shortcomings in the application of all FOUR aspects listed | Approximately 50% of the aspects listed applied correctly | Limited number of aspects listed applied correctly | None of the aspects listed applied correctly | **4** | |
| **Database manipulation** | | | **2** | **1** | **0** | | |
| Delete record(s) and apply changes | | | Meaningfully, appropriately and correctly done<br>Contributes to solution | Mostly meaningful, appropriate and correctly done<br>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** | |
| At least 1 report (output displayed/text file) as a result of processing/data transformation | | | | | | **2** | |
| | | | | | | **50** | |
| **Comments and feedback:** | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| **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 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** (verified during interview)<br>• Test for valid data<br>• Test for extreme data<br>• 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** | |
| **Comments and feedback:** | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| **Learner name** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Task 10** | **4** | **3** | **2** | **1** | **0** | | |
| **Comments/Notes** | Code clearly annotated to fully explain all necessary parts.<br>Extensive project notes present and of an excellent standard. Clearly explains working of the program | Code clearly annotated to explain all necessary parts.<br>Project notes present and of a very good quality | Code annotated to explain most necessary parts.<br>Project notes present and of a moderate standard | Code annotated to explain certain parts.<br>Inadequate project notes present | No comments or no project notes | **4** | |
| **Does the program meet the requirements?** | Exceeds requirements stated in Phase 1.<br>Comprehensive program.<br>All elements function as specified.<br>Shows insight in all aspects. | Meets the requirements stated in Phase 1.<br>Less comprehensive.<br>All elements function as specified.<br>Shows insight in most aspects. | Meets most of the requirements, but some don't function well<br>Only some program elements function as specified in Phase 1.<br>Shows insight in one or two aspects. | Only meets some requirements, and some don't function well.<br>Basic program.<br>Basic scope.<br>Very limited insight. | Does not meet the requirements.<br>Less than basic.<br>Limited scope. | **4** | |
| **Interview** | **8** | **6** | **4** | **2** | **0** | | |
| **Explain selected code** | Explained all selected code clearly and with confidence.<br>Shows excellent insight. | Explained selected code with minor shortcomings.<br>Shows insight | Unable to explain some of the selected code adequately.<br>Shows some insight | Unable to explain most of the selected code, limited insight. | Unable to explain any selected code, no insight. | **8** | |
| | | | | | | **16** | |
| **Comments and feedback:** | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

**Assessment Summary**

| Phase | Tasks | Task description | Maximum Mark | Mark Obtained | Moderated Mark |
|---|---|---|---|---|---|
| **Phase 1** | Task 1 | Task definition and user story and acceptance test | 12 | | |
| | Task 2 | Data dictionary: Arrays, Text files and User defined methods | 16 | | |
| | Task 3 | Data dictionary: Database design | 12 | | |
| | Task 4 | Navigation / flow between screens and GUI design | 4 | | |
| | Task 5 | IPO table and data validation | 20 | | |
| **Phase 2** | Task 6 | Create GUI application | 6 | | |
| | Task 7 | Create a Database and connect to application | 6 | | |
| | Task 8 | Coding | 50 | | |
| | Task 9 | Testing and data validation | 8 | | |
| **Phase 3** | Task 10 | Documentation and Interview | 16 | | |
| | | **Final mark** | **150** | | |

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:**

# 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: _____

Department Head signature: _____ Date: _____

Teacher Moderator signature*[1]: _____ Date: _____

DSA signature: _____ Date: _____

* [1] Lead Teacher / Cluster leader

# 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 | ___ / ___ / 2024 Date |

# Annexure C

## Declaration of authenticity

| | | | |
|---|---|---|---|
| **Learner name** | | **ID Number** | |
| **Grade** | 11 | **Year** | 2024 |
| **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.

_____ ___ / ___ / 2024

Signature of Learner Date