

**Information Technology**

**Guidelines for**

**Practical Assessment Task (PAT)**

**Grade 11**

**2021**

**These guidelines consist of 29 pages.**

**Contents**

[What is the PAT? 3](#_Toc33697091)

[Mark allocation 4](#_Toc33697092)

[The Topic 5](#_Toc33697093)

[What you need to be able to do the PAT 6](#_Toc33697094)

[Malpractice 6](#_Toc33697095)

[Non-compliance 6](#_Toc33697096)

[PAT requirements 7](#_Toc33697097)

[Instructions for Phase 1 8](#_Toc33697098)

[Instructions for Phase 2 – Coding and Testing 11](#_Toc33697099)

[Annexure A: Assessment Tools 14](#_Toc33697100)

[Annexure B 22](#_Toc33697101)

[Learner declaration – Phase \_\_\_\_ 22](#_Toc33697102)

[Annexure C 23](#_Toc33697103)

[Declaration of authenticity 23](#_Toc33697104)

[Guidelines for teachers to provide guidance 24](#_Toc33697105)

[What are the learners required to do and provide? 24](#_Toc33697106)

[How will learners go about it? 24](#_Toc33697107)

[Skills required 24](#_Toc33697108)

[What must the learners be taught beforehand? 24](#_Toc33697109)

[Malpractice 25](#_Toc33697110)

[Learner declaration of authenticity of the PAT 25](#_Toc33697111)

[Role of the teacher 26](#_Toc33697112)

[Supervised/Controlled conditions 26](#_Toc33697113)

[Managing the PAT 27](#_Toc33697114)

[Evidence of assessment 27](#_Toc33697115)

[Interview 27](#_Toc33697116)

[Requirements 28](#_Toc33697117)

[Non-compliance 28](#_Toc33697118)

**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 **TWO phases**, as explained below:

Phase 1: Outlines the project task, solution and a possible design of the project.

Phase 2: A working, fully documented Delphi application that implements the planned solution.

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

**Phase 1: Not later than the first week of of Term 3**

**Phase 2: Not later than the first week of of Term 4.**

LEARNERS NEED TO STRICTLY ADHERE TO THE DUE DATES FOR EACH PHASE.

**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.

|  |  |  |
| --- | --- | --- |
| **Phase** | **Development phase** | **Maximum Mark** |
| **Phase 1** | Analysis and Design | 48 |
| **Phase 2** | Coding and Testing | 64 |
| **General** | Final product and impression | 18 |
| **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.

**The Topic**

**Encryption/Decryption**

Sometimes one wants to ensure that other people cannot read one’s correspondence or messages or that personal information remains confidential. One way of keeping written information confidential or secret is to encrypt the message using a cipher.

Some people, on the other hand, may like the art of cryptanalysis or the challenge to crack ciphers.

You need to develop a program that encrypts messages using a cipher and decrypts messages using the same cipher. For example, if someone sends an encrypted message to another person, that person must be able to use the program to decrypt the message.

You could also have a cipher game where others are challenged to crack the ciphers.

The program needs to use at least two different ciphers:

* **Use at least one existing cipher (see list of examples below)**
* **Use at least one cipher that you developed yourself**

Examples of existing ciphers for encrypting/decrypting messages:

Various examples: <http://www.simonsingh.net/The_Black_Chamber/chamberguide.html>

Vigenere Cipher: <http://sharkysoft.com/misc/vigenere/>

RSA Cipher: <http://cisnet.baruch.cuny.edu/holowczak/classes/9444/rsademo/>

Keyword Cipher: <http://www.secretcodebreaker.com/keyword.html>

The hobby and art of cryptanalysis -- that is, learning to break ciphers, see <http://cryptogram.org/>

Cracking ciphers: <http://cryptogram.org/solve_cipher.html> or <http://simonsingh.net/cryptography/cipher-challenge/the-ciphertexts/>

Ideas for developing your own cipher, using

* **Binary numbers**
* **ASCII codes /symbols**
* **Standardised numbers such as ID numbers, ISBN numbers**
* **Calculations**
* **Mathematical processes such as check digits, LCM, etc.**
* **String processes**
* **Combining aspects of existing cipers, etc.**

**You could expand on the programme that you developed in Grade 10 in 2020 to include the additional requirements for Grade 11.**

Your final program must comprise ***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.

You are not limited to the above-mentioned list of ideas, but you need to keep within the overall theme – encryption. 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.

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

**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)
* An office suite with the following software:
  + Word processing software.
  + Database software.
* Storage media to save and backup your work electronically, for example a flash drive, rewritable CD/DVD.

**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 seriousness of the offence.

**NOTE:** If you use work from other resources, it may not exceed 10% of the work that you submit.

**Non-compliance**

You will be given up to a part of term three 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.

**PAT requirements**

The project must include the following:

* 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 **AND/OR** a database connection and database manipulation that entail performing different CRUD (Create, Read, Update and Delete) operations.
* Other data structures that will be relevant to your program

**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/or** textfile as well as other data structures (e.g. arrays) to provide the necessary input, processing and output
* comply with relevant HCI principles

**Database (Optional)**

When a database is used, the database must:

* Have at least TWO tables (relational tables implementing referential integrity)
* contains sufficient data volumes and uses a variety of field types(approximately 5 fields and at least 10 records per table)
* Be accessed and manipulated by the program using code constructs

**Text files**

The use of a text file(s) for e.g., 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 for Phase 1**

During this phase you have to show that you have done a proper and thorough user requirement analysis. This needs to be done in order 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:

### Scenario and Scope: Define the Task (Page 3 of Word document format)

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.**

### User Requirements (Page 4 of Word document format)

The ***user*** is the target audience and will thus determine the needs and requirements of the program. Determine the clients/users and their requirements.

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

### Navigation/Description of Flow Diagram (Page 5 of Word document format)

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

### Design the Database (Optional) (Page 5 of Word document format)

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

.

Show the design of the database(s), 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.

### Design the Graphical User Interface (GUI) (Page 6 of Word document format)

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 has to 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 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.

### Data Dictionary (Page 7 of Word document format)

**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.

**Database (optional):**

Your application must use a database. Explain how a database can be used in your application so that it adds value to the application.

**Other data structures**

**Array(s)**

Your application must use an array. Explain where a text file can be used in your program so that it adds value to the program.

### Software Tool - Input, Processing, Output (IPO) (Page 8 of Word document format)

### (Format, Data Types/Structures, 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, source of output, validation of input and error checking mechanisms.
* 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.

### Hand In

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

* A clear description of the chosen topic.
* User requirements - detailed information stating the role, activities and limitations of each user of the planned system.
* Data structures including
* A planned database design (Optional)
* The use of a text file(s).
* The use of array(s)
* The use of any other data structure/ advanced programming construct
* A GUI Design.
* The IPO design including validation and error checking techniques.

**Instructions for Phase 2 – Coding and Testing**

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

### Develop the Database (Optional)

Design and construct the database according to the planning document that was developed during Phase 1. 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.

### Develop the GUI

Developing the GUI according to the planning document that was developed during Phase 1. 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.

### Write the Code

Write code to develop the program/system according to the planning document that was developed during Phase 1. 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 multi-nested loops and conditional structures.
* The following data structures are compulsory in addition to the database:
* Text file - reading OR writing OR appending
* Array(s)
* The use of any other data structure not already tested/ advanced programming construct
* Use relevant validation procedures and components.
* Develop a good GUI and rename relevant components.
* 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 (**Delphi code only**) - optional.

Simple queries: search and display

search against one or more criteria

Complex queries: extract data from two tables/ nested loops

calculating aggregates

further processing of data obtained from the database.

dynamic search

### 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.

### Test the Program/System

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

### 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.
* The teacher can use the mark sheet for Phase 2 as a guideline and allocate marks accordingly during the demonstration.
* 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 in order to finalise the mark.

**Annexure A: Assessment Tools**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Phase 1: Name of learner:** | | | | | | | |
| **Scenario/Scope**  **(± 200 words)** | **4** | **3** | **2** | **1** | **0** |  |  |
| **Scenario and Scope**   * Topic is clearly stated. * Thorough description of what the problem/task involves (purpose). * Describe a possible solution for the problem/task. * Brief description of the scope. | An excellent presentation of all FOUR points listed | All FOUR points were presented with short comings  OR  A good presentation of THREE points | THREE points were presented with short comings  OR  A good presentation of TWO points | TWO points were presented with short comings  OR  A good attempt to present ONE of the points | Totally inadequate or not applicable  Poor or no coverage of aspects.  No scope or extremely vague and unclear | **4** |  |
| **User Requirements** | **4** | **3** | **2** | **1** | **0** |  |  |
| State WHO the users are.  Role, activity and  limitations of the users  (In table format or a use case diagram) | * Role, activity and limitations of at least TWO different types of users of the system discussed. * Well documented, neat and to the point. | * Minor shortcomings in discussion of role, activity and limitations of at least TWO different types of users of the system. * Documented well but can improve slightly. | * Shortcomings in discussion of role, activity and limitations of users, for example sections left out. * Only ONE user of the system discussed. * Not well documented but still acceptable. | * Major shortcomings in discussion of role, activity and limitations of users. * Only ONE user of the system discussed. * Poorly documented – not acceptable. | * Not done or incorrect or irrelevant. | **4** |  |
| **Navigation/Description of flow diagram** | **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** |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DATA STRUCTURES** | **4** | **3** | **2** | **1** | | | **0** |  |  |
| **Database design (optional)**   * All fields are relevant * Type and size of fields well-chosen | All database design requirements met. A well designed database. | Good database design with minor shortcomings. | Average design with several shortcomings | Database design done, but with limited value. | | | No database or incorrect or irrelevant. | **4** |  |
| **Text files(s)** | Excellent and relevant description of use of a text file(s). | Good description of use of a text file | Relevant description of use of a text file with some shortcomings | An attempt to describe the use of a text file with many shortcomings. | | | Not done or incorrect or irrelevant. | **4** |  |
| **Array/(s)** | Excellent and relevant description of use of an array. | Good description of use of an array. | Relevant description of use of an array | An attempt to describe the use of an array with shortcomings. | | | Not done or incorrect or irrelevant. | **4** |  |
| **GUI Design** | **4** | **3** | **2** | **1** | | | **0** |  |  |
| * Design fits to program's intended use * Appropriate components * Ease of use, logical flow * Clearly marked navigation * Friendly dialogue/Help | Good GUI design, all of the listed principles applied throughout the system e.g. with data capturing, output, navigation, etc. | Satisfactory GUI design, most (at least 4) of the principles applied throughout the system e.g. with data capturing, output, navigation, etc. | Limited GUI design, most (at least 3) of the principles applied throughout the system e.g. with data capturing, output, navigation, etc. | | Poor GUI design applied less than 50% (less than 3) of the principles. | GUI design is not functional or does not support the intended use at all. | | **4** |  |
| **Data Input** | **4** | **3** | **2** | | **1** | **0** | |  |  |
| **Input interfaces (at least TWO)**   * Source of input, such as from the keyboard, text file, array or the database. * Data type * Format of the input for example date, gender (M/F). * 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  OR  Major shortcomings in describing all inputs according to all FOUR points listed | | Poor attempt to describe input values | No inputs described  OR  Incorrect | | **4** |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input Validation**   * At least THREE different data types validated * At least THREE inputs validated including: * Validate for NULL/empty field AND * Test if value was selected in a selection component * Associated error messages | Clearly describes all points listed | Clearly describes TWO points listed  OR  Minor shortcomings in describing all points listed | Clearly describes ONE points listed  OR  Major shortcomings in describing all points listed | Poor attempt to describe validation | No validation described  OR  Incorrect | **4** |  |
| **Data Processing** | **4** | **3** | **2** | **1** | **0** |  |  |
| **WHAT** processing will need to be done | Clearly lists at least SIX processes to be done. | One or two processes not listed. | About 50% of the processes listed. | Only one or two processes listed. | No processes listed | **4** |  |
| **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/ incorrect or irrelevant | **4** |  |
| **Data Output** | **4** | **3** | **2** | **1** | **0** |  |  |
| **Output interfaces (at least TWO)**   * Data to output * Format of the output, for example currency, date. * 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  OR  Limited outputs described | Poor attempt to describe outputs | No output described  OR  Incorrect | **4** |  |
| **TOTAL** | | | | | | **48** |  |

|  |
| --- |
| **Comments/feedback:**      **Teacher name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Teacher signature: Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Phase 2: Name of learner:** | | | | | | | |
| **Project Assessment** | | | | | | | |
|  | | | | | | | |
| **Database Design (optional)** | **4** | **3** | **2** | **1** | **0** |  |  |
| **Implementation of Database design** | Database design correctly implemented, with at least 2 tables, suitable fields, data types and sizes.  Large/adequate data volume. | Database design correctly implemented, with at least 2 tables, suitable fields, data types and sizes.  Limited volume of data used. | Database design using at least 2 tables, but not properly implemented. Errors in fields, data types and sizes | One table with suitable fields, data types and sizes. | Totally inappropriate or incorrect or not used | **4** |  |
| **GUI Design** | **4** | **3** | **2** | **1** | **0** |  |  |
| **Ease of use / HCI principles**   * Excellent layout and communication (screen tips, feedback, help et cetera) * Most appropriate components * Readable/Relevant   Input/output   * Excellent use of effects/ colour/ icons/ shortcuts/ tool tip text, et cetera | 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** |  |
| **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** |  |
| **Input** | **4** | **3** | **2** | **1** | **0** |  |  |
| **Input data**   * Variety of sources of input, such as from the keyboard, text file or the database. * Correct data types * Appropriate format used for example date, gender (M/F). * GUI component used. | Excellent application of all FOUR aspects listed. | Minor shortcomings in the application of all FOUR aspects listed | THREE of the aspects listed are correctly applied. | Limited application (less than 3) of the aspects listed. | No application of the aspects listed. | **4** |  |
| **Validation/Error catching** | A variety of validation/ error catching for relevant input. Clear and appropriate error messages and exception handling mechanisms. | A variety of 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/catch errors. | **4** |  |
| **Processing** | **4** | **3** | **2** | **1** | **0** |  |  |
| **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** |  |
| **Algorithm efficiency** | All algorithms provide the most efficient solutions.  Good programming techniques used. Effective modular design with correct use of own functions and procedures | Most algorithms provide the most efficient solutions.  Acceptable programming techniques used. Limited modular design with correct use of own functions and procedures | Limited efficiency of algorithms used.  Few algorithms use good programming techniques.  Poor modularity with limited use of own functions and procedures | Poor efficiency of algorithms used.  Algorithms do not use good programming techniques.  Attempted use of own functions and procedures | Totally inadequate or not working correctly | **4** |  |
| **Relevant and appropriate use of complex code for example**  Dynamic component | Excellent use of complex code that works correctly, adds value to the system. | Works correctly, 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** | **4** | **3** | **2** | **1** | **0** |  |  |
| * Layout * Readability/Clarity for eg. columns, headings * Formatted for eg. currency * 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 amount of aspects listed applied correctly. | None of the aspects listed are applied correctly. | **4** |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Database manipulation (Delphi code only) - optional** | | | | **2** | **1** | | **0** | |  |  | | |
| Sort records in a table | | | | Works correctly and is  applicable to task | Works but poorly constructed /  not applicable to task | | Not done, does not work or incorrect | | **2** |  | | |
| Search for data in a table | | | | **2** |  | | |
| Insert a new record to a table | | | | **2** |  | | |
| Delete a record from a table | | | | **2** |  | | |
| Edit selected fields in a record | | | | **2** |  | | |
| String manipulation for example using Left, Mid and Right | | | | **2** |  | | |
| Manipulation using to different calculations such as minimum, maximum, sum and average | | | | **2** |  | | |
| At least one query involving two tables using Delphi code | | | | **2** |  | | |
| **Documentation** | **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 good standard | 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** |  |

|  |  |  |
| --- | --- | --- |
| **Total (implementation):** | **64** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **General: Final product and impression Candidate name:** | | | | | | |
| **Aspect** | **4** | **3** | **2** | **1** | **0** |  | **Mark** |
| **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** |  |
| **Creativity and originality** | Product shows exceptional creativity and originality. Contains feature(s) that are NOT part of the curriculum such as connecting or running on a mobile device. Feature(s) must show a high level of complexity to implement. Learner must show knowledge and skills on how the feature(s) were coded. | Product shows a high level of creativity and originality. Uses eye catching features such as animation using fairly complex code in an original and sensible way to enhance the look and feel/functionality of the product. Learner must show knowledge and skills on how the feature(s) were coded. | Product shows a standard level of creativity and originality. Uses standard Delphi GUI features such as eye catching buttons, animation, etc in an original and sensible way to enhance the look and feel/functionality of the product. Learner must show knowledge and skills on how the feature(s) were coded. | Product shows limited creativity and originality. At least two attempts to apply standard Delphi GUI features to enhance the look and feel/functionality of the product. Learner must show knowledge and skills on how the feature(s) were coded. | Total lack of creativity and originality. | **4** |  |
| **Ability to explain 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 | **4** |  |
| **Attitude and commitment** |  |  | Kept to due dates. Well-designed phases.  Showed exceptional commitment and pride in work done. | One of the phases were not on time.  Showed some commitment and pride in work done. | None of the phases on time. Showed no commitment. | **2** |  |
| **Total:** | | | | | | **18** |  |

**Assessment Summary (with database)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Focus** | **Maximum Mark** | **Mark Obtained** |
| Phase 1 | Analysis and Design | 48 |  |
| Phase 2 | Coding and Implementation | 64 |  |
| General | Final product and impression | 18 |  |
| **Total** | | **130** |  |
| Final mark | |  |  |

**OR**

**Assessment Summary (without database)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Focus** | **Maximum Mark** | **Mark Obtained** |
| Phase 1 | Analysis and Design | 44 |  |
| Phase 2 | Coding and Implementation | 44 |  |
| General | Final product and impression | 18 |  |
| **Total** | | **106** |  |
| Final mark | |  |  |

**Declaration of Authenticity**

|  |
| --- |
| 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:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

**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): |
|  |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ / \_\_\_ / 2021  Signature of Learner Date | |

**Annexure C**

**Declaration of authenticity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Learner name** |  | **ID Number** |  |
| **Grade** | 11 | **Year** | 2021 |
| **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. | | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ / \_\_\_ / 2021  Signature of Learner Date | | | |

# 

# Guidelines for teachers to provide guidance

**What are the learners required to do and provide?**

Learners are required, with appropriate supervision, to:

* Choose an area of interest within the topic/scenario provided
* Formulate a focus question that can be investigated/researched
* Plan, research and carry out the project
* Deliver a report to a specified audience
* Provide evidence of all stages of the project for assessment

**How will learners go about it?**

The learner will:

* Plan and complete an individual project, applying a range of programming and software engineering skills and strategies to meet the objectives as set out by the PAT requirements
* Identify questions to ask
* Obtain, critically select and use selected information from a range of sources; process and analyse data, apply it relevantly and demonstrate understanding of appropriate linkages, connections and complexities of the topic and focus question
* Select and use a range of skills, including design tools and algorithms, solve problems, take decisions critically, creatively and flexibly, to produce a software solution
* Evaluate outcomes both in relation to PAT requirements and own learning and performance.
* Use appropriate communication skills and media to present evidence in appropriate format.

**Skills required**

The learner must be able to:

* Do a complete user requirement analysis which includes a complete description of the role, activities, requirements and limitations of at least TWO different users of the planned system.
* Bring together information to suit the content and purpose
* Apply decision-making and problem-solving skills
* Extend planning, research, critical thinking, analysis, synthesis, evaluation and presentation skills
* Develop confidence in applying the content, programming and software engineering principles and techniques they have studied
* Develop and apply skills creatively, demonstrating initiative and enterprise
* Seek advice and support when needed

**What must the learners be taught beforehand?**

The taught elements include:

* Application software and ICT skills that will enhance the production of the report and the development of the project covering analysis and execution
* Solution development content and skills, including the ability to define a task
* Project management skills, including time, resource and task management

# Malpractice

Learners may NOT:

* Get help/guidance from others without acknowledgment (complete **Annexure B** for EACH phase)
* Allow others to do the programming code for their project
* Submit work which is not their own
* Lend work to other learners
* Allow other learners access to, or the use of, their own independently-sourced source material (this does not mean that candidates may not lend their books to another candidate, but candidates should be prevented from plagiarising other learners' research)
* Include work copied directly from books, the Internet or other sources without acknowledgement and attribution
* Submit work typed or word-processed by another person

These actions constitute malpractice, for which a penalty will be applied.

If malpractice is identified, the assessment authorities must be notified and details of any work which is not the learner's own must be recorded.

# Learner declaration of authenticity of the PAT

For each phase, learners complete a declaration (**Annexure B**) for the work done during that specific phase. All substantive advice/help given to the learners should be recorded as part of the phase documents.

After completing the PAT, learners should sign the declaration of authenticity (**Annexure C**) to confirm that the work submitted is their own.

# Role of the teacher

The teacher will teach the information management content, skills and strategies prior to the project.

While managing the project and supervising the learners, the teacher will:

* Conduct an initial planning review to discuss the topic/scenario, requirements, objectives and development of the project
* Agree on the focus question (learners should record the guidance given as part of the   
  Phase 1 documents, for example where appropriate, record their own initial question with clear evidence of the guidance and the final question)
* Give regular feedback to learners, for example to formulate a focus question that is suitable and manageable
* Assess the work of the learners at the end of each phase using the standardised assessment tool and record feedback given
* Endorse each learner's assessment by signing the assessment tools for each phase including a final declaration that the evidence submitted for assessment is the unaided work of the learner
* Confirm their evaluation based on continuous observation and feedback as well as aninterview session to provide a final judgement regarding independent work, insight and problem-solving
* Make the assessment of the work of the learners following any standardising and internal moderation procedures required

The teacher will assess the potential project (task definition and scope) against the following checklist.

* Is the focus area, suitable for the project?
* Does the focus allow the learner to investigate and to access the higher-level concepts and skills in the assessment objectives, for exampleto plan, research, analyse, evaluate and explain, rather than simply describe and narrate?
* Are the focus question and proposed action clear and focused on an issue which can be managed within the timeframe and available resources?
* Do the focus and proposed action indicate that the learner will be capable of investigating and researching the topic and carrying out the activity or task independently and within appropriate ethical or methodological guidelines?
* Is the learner likely to face difficulties understanding the task and issues associated with the focus?

The teacher will authenticate the PAT:

* Teacher will confirm on the assessment tool thatthe work assessed is solely that of the learner concerned and was conducted under supervised/controlled conditions
* Teacher will sign the assessment tool of each phase

# Supervised/Controlled conditions

The PAT must be managed in such a manner to be able to confirm that the work assessed is solely that of the learner concerned.

# Managing the PAT

The teacher must plan his/her work schedule according to the time allocated for the PAT in the CAPS document for Information Technology (teaching plan for Grade 12).

There are different possible approaches to managing the PAT:

**Option 1:**

* The teacher could dedicate a portion of the time on a weekly basis to the PAT while simultaneously continuing with normal teaching to complete the Grade 12 curriculum in the rest of the week.
* If he/she chooses this option, he/she should start with the PAT process towards the end of the first term, completing one phase per term.

**Option 2:**

* The teacher could dedicate a continuous period of time to the PAT, for example the last week(s) of each term, also completing one phase per term.

# Evidence of assessment

Evidence presented for assessment must show how the individual learner has met the assessment objectives and criteria and include the planning, feedback and progress of the project.

The evidence for assessment will include the following:

* The project product, including (content only, without the cover page, table of contents, references, graphics), design documents,final program(fully documented) and other evidence (for each phase)
* The completed learner assessment tool (for each phase)

# Interview

Guidelines for the evaluation of the project:

* Schedule dates and times for demonstrations – allow about 15 minutes per project.
* Take in all the documentation before the demonstration takes place – at least one week in advance – and evaluate the documentation before the demonstration session.
* Learners should demonstrate their projects electronically on the computer.
* During the demonstration session learners should execute test procedures to show that the entire program is working correctly.
* Use the mark sheet for Phase 2 as a guideline and allocate marks accordingly during the demonstration.
* As part of the evaluation, identify random pieces of programming code in the project and ask the learner to explain the purpose and working of the randomly selected code. This is done to ensure that the learner did the coding him- or herself. A similar type of procedure will be followed during moderation. If a learner cannot explain the code used in the project, a mark of zero should be awarded for the project.
* Make sure that the learner hands in the electronic copy of the project that was demonstrated. Use this copy to allocate any outstanding marks in order to finalise the mark.

# Requirements

**(National Protocol for Assessment Grades R–12, Chapter 3)**

Practical Assessment Task components must:

* Comprise assessment tasks that constitute the learners' PAT mark as contemplated in Chapter 4 of the Curriculum and Assessment Policy Statement for IT
* Include a mark awarded for each assessment task (phase), as well as a consolidated mark
* Be guided by assessment components as specified in Chapter 4 of the Curriculum and Assessment Policy Statement for IT
* Be available for monitoring and moderation
* Be evaluated, checked and authenticated by the teacher before being presented as the learner's evidence of performance

# Non-compliance

**(National Protocol for Assessment Grades R–12, Chapter 3)**

The absence of a PAT mark in IT, without a valid reason, will result in the candidate, not being resulted for the subject.

**The candidate will be given up to the first week of Term 3 to submit outstanding work or present himself or herself for the PAT. Should the candidate fail to fulfil the outstanding PAT requirements, such a learner will be awarded a zero ('0') for the PAT component for IT.**

In the event of a learner not complying with the requirements of the PAT, but where a valid reason is provided:

* He or she may be granted another opportunity to be assessed in the assigned tasks, based on a decision by the Head of the assessment body.
* The learner must, within three weeks before the commencement of the final end-of-year examination, submit outstanding work or present himself or herself for the PAT.
* Should the learner fail to fulfil the outstanding PAT requirements, the mark for the PAT component will be omitted and the final mark will be adjusted for promotion purposes in terms of the completed tasks.

Valid reasons in this context include the following:

* Illness, supported by a valid medical certificate, issued by a registered medical practitioner
* Humanitarian reasons, which includes the death of an immediate family member, supported by a death certificate
* The learner appearing in a court hearing, which must be supported by written evidence
* Any other reason as may be accepted as valid by the head of the assessment body or his or her representative