PRACTICAL ASSESSMENT TASK

PHASE 1

[Your Name & Surname Here]

[Your Register Class e.g. 12NG]

TASK DEFINITION

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

**PAT INSTRUCTION**

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.
* The scope of the project is clear and well defined in the format of a simple/brief description of the project.

**NG’s TRANSLATION/NOTES:**

* Start by stating what the problem is.
  + Either fabricate one or choose a real-life problem to work with.
* Next, state the proposed solution to the problem you just mentioned.
  + i.e. you will write a program to do this and that which will help so and so do whatever they have an issue with more easily than before.
  + be careful not put in too much detail here.
* Remember to either explicitly state the scope (i.e. the extent of the program how big you want to go) or include it as a part of your explanation. Scope refers to how big you want to go (small program for one person or a program that is supposed to serve multiple roles for a big company?)
* DO NOT ADD FLUFF!!
  + Do not mention things like “For my Practical Assessment Task…” or “I will have a good looking, easy to use GUI…” or “My program will make use of a database….” or “I will be using Delphi to code…”

**EXAMPLE:**

Johnny owns a vegetable farm up the east coast of South Africa. Despite the drought his farm is doing very well, and he intends to expand his operations. Unfortunately for him, vegetables are notoriously difficult to cultivate. If he wants to increase his output, he will need some way to manage things better.

I intend to code a program for him in which he can catalogue, keep track of, and manage the various vegetables that he grows on his farm. In addition, he will also be able to see statistics based on previous harvests to make calculated decisions regarding what to do in the future. (105 words)

USER REQUIREMENTS

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

**PAT INSTRUCTION**

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.

**NG’s TRANSLATION/NOTES:**

* Your goal here is to put yourself in the shoes of the user and determine exactly
  + what each user will need out of the program
  + who will be using the program
  + what role each user will fulfil.
  + what limitations will be placed on each user
* The simplest way is to do this using a table, though a Use Case Diagram (<https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/what-is-a-use-case-diagram-in-UML/UML_use_case_example-800x707.PNG>) is also acceptable.

**EXAMPLE**

|  |  |  |
| --- | --- | --- |
| **User Role/Who** | **Description** | **Limitations** |
| Bursar  (To keep track of payments) | * View Statements * Issue Bonuses | * May not add new staff or users * May not edit existing staff or users |
| Secretary  (Staff management and bookings) | * Add new staff * Edit staff details * Make appointments | * Cannot view financial statements |
| Administrator  (Manage key parts of the program) | * Add new users * Edit users * View access log * Adjust calculation modifiers * View Statements | * No limitations |

SOFTWARE DIAGRAM

**PAT INSTRUCTION**

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.

**NG’s TRANSLATION/NOTES:**

* Pretty straight forward. Just remember that you need to make sure you have arrows to indicate the direction of the flow.

**EXAMPLE:**

HELP

Will contain detailed info on every function in great detail. Limited help will be available on each form.

MAIN MENU

Serves as central hub for all functions.

CLIENTS

For managing clients to the business. Accessible only to the administrator and the clerk.

STATS

A comprehensive set of tools and graphs to enable the administrator to have a better understanding of the various operational affects and effects of the business.

PRINT

Can print the stats or other information.

DB

Storing all relevant client and sales data.

DATABASE DESIGN

**PAT INSTRUCTION**

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 AND SQL statements.

Show the design of the database, including the tables, relationships, 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, e.g. update/edit/ delete/add data, provide results of queries, provide reports, etc.

**NG’s TRANSLATION/NOTES:**

* Split this section into two parts i.e.
  + DB Design
  + DB Diagram
* In the “DB Design” part, you will design the tables that you will use in your database AND explain how the database will be used as a part of your software.
* Also include an ERD to show the relationships between the tables.

**EXAMPLE OF DB DESIGN**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name: Client | | | |
| Field: | ClientID(PK) | ClientName | AmountOutstanding |
| Data Type: | Text (4) | Text (50) | Currency |
| Example Data: | C001 | Johnny Depp | R55,21 |
| This table has to store all the various details of the business clients. | | | |

**EXAMPLE OF DB DIAGRAM**

Client

Order

DATA DICTIONARY

Class(es)

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

**PAT INSTRUCTION**

**Classes and objects**

Your application must contain at least one object class. Explain where objects can be used in your application so that it adds value to the application.

**NG’s TRANSLATION/NOTES:**

* Create a UML Class Diagram for your class.
* Write a brief paragraph explaining the purpose of your class and how you will be using it.
* Remember: Your class must have meaning! Do not just add a class for the sake of having one – you will lose marks if you do.

**EXAMPLE OF CLASS DIAGRAM AND DESCRIPTION**

|  |
| --- |
| TBus Class |
| - fDriverName : String // name and surname of bus driver  - fDriverCode : String // unique alphanumeric code identifying bus driver  - fNumPassengers : integer // number of passengers currently on bus  - fCapacity : integer // max number of passengers that can be on the bus  - fTicketPrice : real // price that a passenger pays per ticket  - fIsDoubleDecker : boolean |
| + constructor create (driverName, driverCode : String; numPassengers, capacity : integer; ticketPrice : real; isDoubleDecker : boolean)  + setDriverName(driverName: String)  + setDriverCode(driverCode: String)  + getCapacity(): integer  + getNumPassengers(): integer  + calculateTotalTicketPrice: real  + toString() : String |
| The purpose of this class is help manage the busses and tourists that will be riding on them. Each bus in the fleet of the company is instantiated as a separate object to be used in the program. The key method is the “calculateTotalTicketPrice” that determines the amount of income generated based on the number of passengers currently on the bus. |

Text Files

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

**PAT INSTRUCTION**

**Text files**

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

**NG’s TRANSLATION/NOTES:**

* Write a short paragraph as to how you will be using text files as a part of your software. Be descriptive.

**EXAMPLE OF TEXT FILE USAGE**

Based on the Departure and Destination locations, the program will generate an itinerary that will be printed out and stored in a text file. The text file will be named according to the user’s ticket number.

Other Data Structures

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

**PAT INSTRUCTION**

**Other data structures/Advanced programming constructs**

Your program must use a one--/two-dimensional array/an array of objects OR apply programming concepts, such as inheritance, polymorphism, overloaded methods, method binding, etc.

**NG’s TRANSLATION/NOTES:**

* An array is compulsory (1D or 2D) – you can have parallel arrays.
* Inheritance, Polymorphism, Overloaded methods, Method Binding are NOT compulsory. If you would like to know more, come and speak to me.
* You must write a short paragraph on what the array is that you will use and how you will use it.
* Draw up a table of how your array looks. This must clearly show me how many elements the array has, the data type and name of the array.

GUI DESIGN

**PAT INSTRUCTION**

Design a GUI that considers good human-computer-interface (HCI) principles, that prevents errors occurring from invalid input and that minimises the amount of information a user has to input.

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

• The user – type and context

• User requirements/needs, usability

• Dialogue – must be relevant, simple and clear

• Icon usage and presentation – well selected and relevant, well placed with a clear purpose

• Colour – use and combination of colour

• 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 usage of space on the screen

Provide sample(s) of planned data capture and data entry designs (prototype screen dumps may be used but must be annotated) and of planned valid output designs.

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

**NG’s TRANSLATION/NOTES:**

* I have created a table on the next page. You must complete the table for each form/tab/GUI that you have in your software.
* Copy and paste the table below one another for however many times you need it.
* Remember to be consistent across your various forms.
* Don’t forget to clearly indicate how you will be providing help, navigation (closing the form, going back to main menu etc.) as you get marks for this.
* Once again, check the rubric to make sure that you cover everything!

**GUI TEMPLATE**

|  |  |
| --- | --- |
| **Design:** | |
| **Purpose/Use:** |  |
| **Components:** |  |
| **Data Validation:** |  |
| **Additional Info:**  **(Optional)** |  |

IPO

**NG’s TRANSLATION/NOTES:**

I decided not to copy the instructions for the PAT document as it is very lengthy. Make sure you read it first before proceeding. Also look at the rubric for this section. The IPO counts a whopping 20 marks out of the possible 48 marks you can get for Phase 1. Make sure you do this section properly! **CHECK YOUR RUBRIC TOO!**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task:** | **Input** | | | | | | **Processing** | | **Output** | | |
| **General** | | | **Validation** | | |
| **Source (GUI Component)** | **Data Type** | **Format** | **What** | **Method** | **Feedback** | **What** | **How** | **What** | **Format** | **Object** |
| User Registration | Keyboard (edtName) | String | Sentence Case | Numbers/  Special Characters | Loop through and check each char. | “Error! Your name cannot contain….” | Store in variable  Preferably Pseudocode, but Delphi code & SQL accepted. | sName 🡸 edtName.Text | Confirmation Message: “Your details have been stored successfully” | Text | Dialog Box |
| Keyboard (edtSurname) | String | Sentence Case | Numbers/  Special Characters | Loop through and check each char. | “Error! Your surname cannot contain….” | Store in variable | sSurname 🡸 edtSurname.Text |
| Mouse (dtpDOB) | TDate | Regional Settings  (dd/mm/yyyy) | None  (built-in) | N/A | N/A | Store in variable | sDOB 🡸 DateToStr(dtpDOB.Date) |
| Keyboard (edtCell) | String | Digits without any spaces in between e.g.  0789419712 | Has 10 Chars | if(length(edtCell) <10) then…. | “Your cell phone number must have 10 chars” | Store in variable | sCell 🡸 edtCell.Text |
| Calculate Total Price | Mouse (spnQty) | integer | Numeric | Range (no more than 5 items) | MaxValue Property | “Error! You cannot have more than 5 items!” | Calculate Price | quantity \* price (constant) = total price | “Your total comes to <price>” | Text  Currency | Price Label |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task:** | **Input** | | | | | | **Processing** | | **Output** | | |
| **General** | | | **Validation** | | |
| **Source (GUI Component)** | **Data Type** | **Format** | **What** | **Method** | **Feedback** | **What** | **How** | **What** | **Format** | **Object** |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

Annexure A – Declaration of Authenticity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Learner's name** |  | **ID Number** | |  |
| **Grade** | 12 | **Year** | | 2021 |
| **Subject** | Information Technology | | | |
| Practical Assessment Task (PAT) | | **Teacher** | | Mr GHI Nieuwoudt |
| I hereby declare that the contents of this assessment task are my own original work (except for items listed below or 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.  **List of assistance received:** | | | | |
| Nature of assistance | | | Person who provided assistance | |
|  | | |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_ / \_\_\_ / 2021  SIGNATURE OF LEARNER DATE | | | | |