Grade 11 IT PAT Template 2021

**PHASE 1**

* Professional layout
* Headings and front page

1. **Scenario & Scope**

(±200 words)

* Topic clearly stated
* Description of what the problem involves
* Describe your solution for this task
* Brief description of the scope

1. **User Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Users** | **Role** | **Activities** | **Limitations** |
| (user 1) |  |  |  |
| (user 2) |  |  |  |

1. **Navigation/Description of Flow Diagram**

A diagrammatical representation of the design and flow of events when the program is used.

*Google the following sources:*

* *Draw.io*
* *Getafix*
* *Paraben*
* *Creately*
* *Lucidcharts online*
* *yEd graph Editor –* [*www.yworks.com*](http://www.yworks.com/)
* [*http://dia-installer.de/*](http://dia-installer.de/)

1. **Data Structures**

* **Database**: Screen shots of your Database (both Views)
* **Text Files**: Write a short paragraph as to how you will be using text files as a part of your software. Be descriptive.
* **Arrays**: An array is compulsory – you may have parallel arrays. Write a short paragraph on what the array is that you will use and how you will use it. Remember the data type and names of the arrays.

1. **GUI design**

* At least 2 GUI’s ( Tabs / Forms )
* All of the following principles must be in your GUI design:
  + Design fits program’s intended use.
  + Appropriate components
  + Variety of components
  + Logical flow, clear navigation
  + Help available
  + Clear labelling to user on how to use program
  + Reset available where appropriate

1. **Software Design Tool: IPO – ANY TWO (Tab sheets / Forms)**

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | VALIDATION | PROCESSING | OUTPUT |
| -What data will be stored | -What data will be validated | -What will be processed | -What will be displayed |
| -Data type | -Show algorithm | -How will the processing be done | -Format to be displayed |
| -Variety of source: From keyboard / mouse click / computer generated | -Associated error message |  | -Object to be displayed in |

**Examples of IPO’s:**

**Input, processing and output IPO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | **Source** | **Data Type** | **Format of the Input** | **GUI component** | |
| Plain Text | Keyboard | String | Sentence in English | Edit box | |
| **Data Validation** | Range check  sPlainText[i] IN [‘A’..’Z, ‘a..z’, ‘ ‘]  Error message 1  “Only Alphabet letters and spaces allowed” | | | | |
|  | Null Check  sPlainText = ’’  Error message 2  “Edit cannot be empty, provide a message to encrypt” | | | | |
| **Processing** | **Action** | | | | **GUI component** |
| Generate Cipher text | WHAT:  Capture plain text and encrypt to Cypher text  HOW:  Read plain text  Do null check if true show Error message 1  **else**  Do range check if false show error message 2 **else** //encrypt  FOR 1 to the length of the Plain text  Letter  a letter from the plain text  Determine the position of the letter in the alphabet  Add each corresponding letter in the mono alphabet to the cipher text | | | | Button |
| **Output** | **Format** | | | | **GUI component** |
| Cipher Text | String | | | | RichEdit |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 | 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** |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |