**NATIONAL CERTIFICATE (VOCATIONAL) NQF LEVEL 3**

**REVIEWED ICASSPRACTICAL ASSESSMENT**

**TASK 1**

INSTRUCTIONS TO STUDENTS

**SUBJECT: PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3**

**SUBJECT CODE: 10041023**

**DATE OF IMPLEMENTATION: 01 JANUARY 2019** This document consists of 16 pages.

Copyright reserved Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

This is a practical application task that tests your ability to apply the knowledge and skills acquired in the vocational subject. The work required from you is an important and compulsory component of your studies.

**1. General information**

1.1 Study the task carefully.

1.2 Note the mark and time allocations per sub-task and for the task as a whole. 1.3 Note the assessment date(s).

1.4 Study the resources required carefully.

1.5 Use the assessment tool as guidance on how your work will be marked.

1.6 You are required to provide proof that you submitted work done by yourself. This implies that you will be guilty of a malpractice if there is any evidence to suggest that the work submitted is not your own.

As the PAT 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 students.

• 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 that if you use work from other resources, it may not exceed 10% of the work that you submit.

1.7 Save your work regularly. Keep a backup copy of this task. You may need to use this task in practical assessment task 2 and the ISAT task.

**Preamble**

Before you start with this practical assessment task, ensure that you understand and have sufficient practice in all the outcomes of Topic 2 of your curriculum, such as: • Developing and constructing IPO charts and pseudocode as part of the planning process.

• Creating and developing VB.NET GUI based applications that include: o Common controls such as text boxes, labels, list boxes, combo boxes and radio buttons, buttons and events;

o User defined functions and subroutines.

Remember that it is *your responsibility* to practice writing code and solving problems. In doing so you will become a skilled developer.

Copyright reserved 2 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

**2. Description of the task**

You work for a software development company as a programmer. A local company has approached your organisation for their software requirements. You have been assigned to this project. This local company is running a promotion for their products. The promotional products are divided into two groups, namely group A and group B. Customers must choose only one product from group A and zero or more products from group B.

No promotional discount will be given if they no products from group B.

Discount is offered based on the combination of one product chosen from group A and the products chosen from group B. The promotional discount varies depending on the different product mix or depending on what the customer had chosen.

VAT calculated at 15% must be added to the discounted total amount.

The application that you are going to develop must show the following:

• amount before the promotion discount;

• the promotion discount;

• VAT calculated after the discount; and

• the *total amount* the customer will pay, including VAT.

Customers will use this application to try different product combinations before buying their final selection. The application should provide applicable controls for the customers to enter their name and surname, cell phone number and email address.

Customers will typically select a single product from group A and then select products from group B.

When the user clicks on a button (Show Promotional Offer), information regarding the customer details, as well as the promotional offer based on the selected products must be shown.

*Note: You could regard the tasks required for the (Show Promotional Offer) button as generating a quote.*

Your application should include subroutines or functions to provide the following functionality:

• Calculating and returning the applicable discount based on the selected products from group B. *Note: This subroutine will be based on the criteria you have determined or researched with regard to the discounts given based on the group B items.*

• Calculating and returning the VAT amount if the total discounted amount is passed as a parameter.

• Calculating and returning the final total amount that the customer must pay. • Generating a ‘quote’, i.e. showing the details of the promotional offer based on the customer selection.

In addition, your application should be designed in such a way that appropriate communication with the user is ensured.

Copyright reserved 3 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

Your program code should adhere to good programming practices by using: • Applicable variables with the correct scope.

• Correct conversion of datatypes where applicable.

• Correct formatting of output.

• Modular programming approaches that entail the use of functions and subroutines.

**3. Instructions for the task**

**3.1 Sub-task 1: Information gathering and program design** (Total: 5 marks)

This sub-task must be done in your own time of approximately 5 hours in preparation for the main task.

You are required to visit any shop in your area and talk to one of the sales personnel or a manager. As part of your visit, you will present a letter to the owner or manager that has been drafted by the college to inform him/her about the purpose of the visit and that it is for educational purposes. Enquire whether their company is to host a promotion based on the task description and how they would go about in doing so.

Enquire about the typical products that they sell and how various groupings of products could be grouped to allow the customer possible discount. For example:

If the shop you are visiting is a cell-phone dealer then the product listed in group A could be the cell phone itself and the products in group B the accessories. Therefore, the customer must buy a phone to be eligible for discount on the extras such as pouches, memory cards, and other accessories. *Note: You may not use a cell phone dealer as your company of choice.*

Use the interfaces and algorithms created in sub-task 1. Plan design and code the software program following the guidelines below.

3.1.1 Create a word processor document that will allow you to capture information about the company, the different possible products for group A and group B and the possible combinations and discounts, i.e. the promotion details, that could apply. Your document should be presented in a neat and concise manner, with applicable formatting applied.

You should also include the rules that apply for the promotion as well as how and which combinations of products are allowed.

*Note: You only need to gather information for 3-5 different products for group A and 2-5 different combinations of products for group B. (Do not include more than 5 products as it will overcomplicate the exercise.)*

Once you have completed the document, print it and have the owner of the company apply a stamp or his/her signature to the document. Add the company logo or an image of the store to your document. Remember this is an educational exercise and you could include a disclaimer in your document. Your student details are also required. Ensure that you have a digital file of the image or logo. (2)

Copyright reserved 4 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

3.1.2 Based on the promotion details, create a simple flowchart *or* table that will clearly present the exact details of the promotional offer. Using this chart or table, a reader should be able to deduce how the discounts would apply based on the combination of products from group B and the product selected in group A.

Also include a detailed description of the promotional offer, i.e. business rules regarding the promotion and the conditions that apply based on the product combinations. (3)

**3.2 Sub-task 2: GUI planning and logical design** (Total: 13 marks)

This sub-task must be done in your own time of approximately 4 hours in preparation for the main task.

3.2.1 The general main composition of the applications that your software company develops follows the general layout of the design as presented in the wireframe/storyboard below.

Company GUI Layout Template

Company Logo and name

Customer Information

Product Information / Selection

Clear All Show Promotional Offer

Output area

Exit

Notes on the layout:

• The company image or logo is displayed with the company name.

• In the second area, controls should be added to capture the customer details. • The third area should include appropriate controls to have the customer only select one item from group A and one or multiple items from group B.

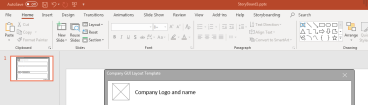
• The fourth area is used for display and output purposes and should include controls that will allow the user to see what s/he would pay based on the deal selected, i.e. the quote.

• Use either Microsoft PowerPoint or another software tool such as ***Draw.io*** to complete the design of your user interface. You could change the headings of the various groupings and your interface design should show the various controls that you are going to use to capture the applicable data and to display the results.

Using a newer version of MS PowerPoint, select the Storyboarding tab sheet to assist you with designing your UI.

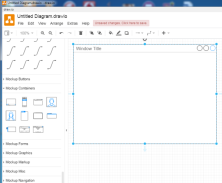
Copyright reserved 5 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019



If your college’s version of PowerPoint is outdated, download a small free application such as Draw.io from https://www.draw.io or use to online version of the tool.

Use the Mock-up stencils to design and draw your UI



As part of your GUI design, indicate the type of the component as well as the names for each. Also write a small single sentence motivation on why you have selected the applicable control.

Copy and paste your GUI designs to a document that includes your student details, assignment details, subject details and college details.

Your lecturer will provide you with a template for this purpose. (5)

3.2.2 Logical planning is an integral part of any software developers’ tasks. *Remember that a program well planned is a program half done!*

For the planning of your application, include the following for each of the subroutines your application will utilise. (Refer to the task description.)

• An IPO chart.

• Definitions and descriptions of the variables and controls to be used. *(Note: Not all subroutines will use or reference controls.)*

• The corresponding pseudocode.

In addition, write the pseudocode for the click event of the (Show Promotional Offer) button.

Label each of your sets of pseudocode appropriately.

Copyright reserved 6 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

Your pseudocode and IPO charts should be presented in a professionally formatted Word document, following your GUI design. (Your lecturer will provide you with a template for this purpose.) (8)

Refer to Annexure 1 of this document for additional content and two examples on this requirement.

Please note that although this part of the work do not count much, it is pivotal for the next stage. Do your planning carefully. Write and complete all your IPOs and sets of pseudocode yourself. The better and more detailed this work is, the easier the coding part will be.

**3.3 Sub-task 3: Coding, development and debugging** (Total: 27 marks; 150 minutes)

This task must be done under supervision. All rules regarding examinations will be applicable.

Use both the documents you have created for sub-tasks 1 and 2 and develop a corresponding VB.NET application.

3.3.1 Create a VB.NET Forms Application and name and save the application appropriately. (1) 3.3.2 Create the GUI with the appropriate controls as you have selected as part of your design.

Provide applicable names for all the controls you utilise and ensure that your GUI are neatly formatted and that your controls are aligned. (3)

3.3.3 Declare all the necessary form class variables. Ensure that you declare your variables with the correct scope and type. Marks will be deducted for incorrect declarations. (3)

3.3.4 Write the code for the corresponding subroutines/functions as you have done in your planning.

Ensure that each of your subroutines receive the correct parameters and return the values where and as necessary. Where best suited, rather pass values as parameters rather than implementing global variables that should be avoided.

• Subroutine used for the discount. (3)

• Subroutine used for VAT. (2)

• Subroutine used for the final amount. (3)

• Subroutine for showing the promotional details. (4)

3.3.5 Include a suitable mechanism and add code to your application to ensure that the user is only allowed to select 1 item from group A. The user should not be allowed to make any selection from the group B items unless one item is selected from the group A items. (2)

3.3.6 Write code that will assign the data entered by the user to the correct corresponding variables. (1)

Copyright reserved 7 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

3.3.7 Write the corresponding code for the click event of the (Show Promotional Offer) button. This button should generate an informative quote based on the items/products selected. Call the applicable subroutines. (2)

3.3.8 Add the code for the (Clear All), and (Exit) application buttons. (1)

3.3.9 Add detailed comments to your code including your student details. (1) 3.3.10 Include any additional functionality to illustrate how the application could be enhanced. (1)

**3.4 Sub-task 4: Demonstration, presentation and documentation** (Total: 5 marks; 30 minutes)

This sub-task must be done under supervision following sub-task 3.

The sub-task involves documenting the program by adding comments to the code, printing and submitting the software evidence. Follow the guidelines below to complete the sub task.

3.4.1 Create a professionally formatted document that includes the documents of the previous stages and screenshots of your application in the VB.NET IDE as well as of your application running with different sample data. (3)

3.4.2 Present your application to your lecturer. For this task, you need to be knowledgeable of the code you have written as well as provide motivations regarding your design. (2)

3.4.3 Submit a soft copy of the program to your lecturer at the end of the demonstration. **4. Time and mark allocation**

| **Sub-task** | **Activity** | **Time**  **allocation** | **Mark**  **allocation** | **Assessment tools** |
| --- | --- | --- | --- | --- |
| 1 | Information gathering |  | 5 | Checklist |
| 2 | GUI planning and logical design |  | 13 | Checklist |
| 3 | Developing, coding and debugging. (Supervised) | 150 minutes | 27 | Checklist |
| 4 | Documentation, presentation and submission. (Supervised) | 30 minutes | 5 | Checklist |
| **TOTAL** | | **180 minutes** | **50** |  |

**5. Assessment tool**

Use the marking checklist on page 9 as guidance on how your work will be marked. The mark allocation is a guide to assess each sub-task. Your response will be assessed in terms of the quality, appropriateness, and level of effort/analysis you have put in.

Copyright reserved 8 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

| **STUDENT’S CHECKLIST PRACTICAL ASSESSMENT TASK 1** | | |
| --- | --- | --- |
| **Task description**  Promotional quote software application. | **SECTION MARK** | **TICKLIST** |
| **Sub-task 1: Information gathering and program design** | **[5]** |  |
| 3.1.1 *Electronic document for information capturing – criteria:*  • A formatted cover page including the student’s details.  • An applicable company image or logo included.  • Presentation contained detailed descriptions of at least 3 products listed as part of group A. Each product was presented with a name and price and applicable products were listed as part of group B.  • Presentation contained detailed descriptions of the applicable products and the discount information for group B.  • Disclaimer document included and signed/stamped.  (2 = all the criteria are met; 1 = one or two criterions is not met; 0 = more than two criterions are not met.) | 2 |  |
| 3.1.2 *Flowchart/table of promotional offer and accompanying description – criteria:* • A well-presented flowchart/table to represent information regarding the applicable discount per product as part of group B, based on the selection of the group A product. An applicable company image or logo included.  Flowchart/table contained detailed descriptions of at least 2 products for group B, as well as the relationship and applicable products matching group A. Each of these products was presented with a name, price and applicable promotional information. The presentation contained detailed descriptions of the applicable products and the discount information for group B.  • The promotional offer details were adequately described and was easily interpretable.  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |
| **Sub-task 2: GUI planning and logical design** | **[13]** |  |
| 3.2.1 *GUI design and composition – criteria:*  • A well formatted document including all the required details as stipulated are presented  • A detailed wireframe created with a software tool are presented. The wireframe indicates the various components and are grouped in the given format as presented. The wireframe is presented neatly with the controls aligned and suitable for its purpose  • The wireframe contains descriptions for each component and a suitable name • All controls selected and included are motivated.  (5 = all the criteria are met; 3 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 5 |  |

Copyright reserved 9 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| 3.2.2 *Logical Planning – criteria:*  • All *four* IPO charts were created with the input variables, basic algorithm, and output variables identified. The algorithms were correct in relation to the variables and the task at hand.  • Each IPO (i.e. all four) had a corresponding description of the input, possible intermediate, possible constants and output variables identified.  • For each of the variables (for all of the IPOs) a suitable datatype and name was given. Where applicable a suitable component was also indicated.  • The pseudocode evaluated were presented in a readable way with clear indications of the logical flows and constructs required.  • The appropriate input and processing and output operations (or in the case of a subroutine, the parameters being passed) were identified and used. • The pseudocode (all four sets) related to the algorithm and were logically correct. • The code was indented and presented neatly with comments where applicable. • A well-formatted document including all the required details as stipulated were presented.  • The content of the logical planning document was neatly presented and there was evidence of pride and professionalism in the presentation thereof.  (8 = all the criteria are met; 4 = one criterion is not met; 2 = two criterions are not met; 0 = more than two criterions are not met.) | 8 |  |
| **Sub-task 3: Coding, development and debugging** | **[27]** |  |
| 3.3.1 *VB.NET forms application created and saved – criteria:*  • The project and corresponding VB files suitably named and saved. (1 mark) | 1 |  |
| 3.3.*2 Creation of the GUI, placement and naming of controls – criteria:* • The given GUI outline was used.  • The GUI design completely correlated with the wireframe.  • The design was neatly formatted, uniform and the component placement conformed to good UI design practices.  • The image or company logo displayed or loaded when the application runs. (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |
| 3.3.*3 Declaration of variables – criteria:*  • All applicable Input, Intermediate and Output variables are suitably declared • All applicable Input, Intermediate and Output variables are suitably initialised • All variables are declared with the correct scope  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |
| 3*.3.4 Coding and corresponding subroutines – criteria:*  *Subroutine for discount*  • The subroutine had all the necessary parameters.  • The subroutine will function according to scope and the logical constructs required, i.e. it will work correctly.  • The coding related to the pseudocode and planning.  • Parameters were declared and used correctly.  • Where applicable, the correct value returned, or reference parameters used. (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.)  *Subroutine for VAT*  • The subroutine had all the necessary parameters.  • The subroutine will function according to scope and the logical constructs required, i.e. it will work correctly.  • The coding related to the pseudocode and planning.  • Parameters were declared and used correctly.  • Where applicable the correct value returned, or reference parameters used. (2 = All the criteria are met; 1 = One criterion is not met; 0 = More than one criterion is not met.) | 3  2 |  |

Copyright reserved 10 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| *Subroutine used for the final amount*  • The subroutine had all the necessary parameters.  • The subroutine will function according to scope and the logical constructs required, i.e. it will work correctly.  • The coding related to the pseudocode and planning.  • Parameters were declared and used correctly.  • Where applicable, the correct value returned, or reference parameters used. (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.)  *Subroutine for showing the promotional details*  • The subroutine had all the necessary parameters.  • The construction of the subroutine and the flow thereof to enable the user to see the various promotional details, was correct and functional.  • The subroutine will function according to scope and the logical constructs required, i.e. it will work correctly.  • All product details were correct.  • The coding related to the pseudocode and planning.  • Parameters were declared and used correctly.  • Where applicable, the correct value returned or reference parameters used. (4 = all the criteria are met; 2 = one criterion is not met; 0 = more than one criterion is not met.) | 3  4 |  |
| *3.3.5 Mechanism to control item selection – criteria:*  • The application included code or a mechanism to allow the user to only select one item from group A.  • No items from group B may be selected unless an item from group A was not selected first.  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| *3.3.6 Assignment of input data – criteria:*  • All the input variables were appropriately assigned.  • All conversions were correctly done from the input controls.  (1 = all the criteria are met; 0 = one criterion is not met.) | 2 |  |
| *3.3.7 On click event of the show promotional offer button – criteria:* • Clicking on the button resulted in an informative output message being displayed in the output area.  • The message was appropriately formatted.  • The coding applied correlated to the logic requirements as identified, with the appropriate function calls.  • The logical composition and flow of the code presented was formatted and showed that the student mastered the understanding of modular programming. (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| *3.3.8 Clear all and exit buttons – criteria:*  • Clicking on the *clear* button resulted in all the input fields being reset and cleared to the initial state of the form.  • Correct methods were applied.  • Clicking on the *exit* button exited the application.  • The user will confirm the choice made to exit.  (1 = all the criteria are met; 0 = one criterion is not met.) | 1 |  |
| *3.3.9 Comments*  Detailed comments were included throughout the code of the application and contributed to the understanding and readability of the code.  (1 mark) | 1 |  |

Copyright reserved 11 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| *3.3.10 Additional functionality and enhancements – criteria:*  • Student was able to identify and explain two areas in the application code where additional features and enhancements were added.  • Each area contained at least 2-3 logical code constructs.  (1 = all the criteria are met; 0 = one criterion is not met.) | 1 |  |
| **Sub-task 4: Demonstration, presentation and documentation** | **[5]** |  |
| *3.4.1 Application document – criteria:*  • The created document was based on a template and included all the various elements of the previous sub-tasks.  • Uniform fonts were applied and the document had a professional appeal. • The document contained a cover page with the appropriate details. • An index was included.  • Applicable running screenshots with descriptions were included as part of the user documentation.  • There was evidence that the student took pride in his/her work and the document could be used for reference purposes by future students.  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |
| *3.4.2 Demonstration – criteria:*  • The student comfortably demonstrates the application and are knowledgeable on all the code constructs implemented by answering all questions correctly. • The application works perfectly as intended  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| **TOTAL MARKS** | **50** |  |

Copyright reserved 12 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

| **6. RECORD OF PERFORMANCE IN ICASS PRACTICAL ASSESSMENT TASK 1** | | | |
| --- | --- | --- | --- |
| **College** |  | | |
| **Campus** |  | | |
| **Student’s Surname and First Name/s** |  | | |
| **Student’s ID Number** |  | | |
| **Lecturer’s Surname and Initials** |  | | |
| **Date of conclusion of assessment** |  | | |
| **ASSESSMENT GRID** | | | |
| **SUB-TASKS** | | **MARK ALLOCATION** | **STUDENT’S MARK** |
| Sub-task 1: Information gathering and program design | | 5 |  |
| Sub-task 2: GUI planning and logical design | | 13 |  |
| Sub-task 3: Coding, development and debugging | | 27 |  |
| Sub-task 4: Demonstration, presentation and documentation | | 5 |  |
| **TOTAL** | | **50** |  |
| **Converted to %** | | **100%** |  |

| **COMPETENCE LEVEL INDICATORS** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Outstanding** | **Highly**  **competent** | | **Competent** | **Not yet competent** | **Not achieved** |
| **80-100%** | **70-79%** | | **50-69%** | **40-49%** | **0-39%** |
| **5** | **4** | | **3** | **2** | **1** |
| **Student’s Competence Level** | |  | | | |
| **Student’s Signature** | |  | | | |
| **Lecturer’s Signature** | |  | | | |
| **Date** | |  | | | |
| **Moderator’s Signature** | |  | | | |
| **Feedback to student** | | | | | |

Copyright reserved 13 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

**Annexure A**

Examples of IPO’s and corresponding pseudocode and Code in VB.NET **Example 1**

**Problem statement**

William bought a new house with a swimming pool for which the dimensions are 9 m × 7 m. He needs to install paving as well as fencing around the pool. He would like to determine the cost. Provide the following values:

• Width of paving around the pool.

• Cost of paving per square meter.

• Width of garden around the paving.

• Cost of fencing per meter.

Calculate and display the price he has to pay for the paving and for the fencing.

Display both answers in a user-friendly way. You may assume that all input will be valid. **IPO**

| I | P | O |
| --- | --- | --- |
| widthPaving  widthGarden  costPaving  costFencing | Prompt to read input values  Enter input values  Calculate price that paving will cost Calculate price that fencing will cost Display answers | pricePaving  priceFencing |

**Definitions and descriptions of the variables and controls to be used**

**Description Type Name of variable Control Input:** Width of the paving around the pool real widthPaving Text box Width of the garden around the paving real widthGarden Text box Cost of paving per square meter real costPaving Text box Cost of fencing per meter real costFencing Text box **Intermediate**: Total area of pool and paving around it real totalArea

Area of paving (total area – pool area) real areaPaving

Total length of paving (needed to calculate area) real totalLengthPaving

Total width of paving (needed to calculate area) real totalWidthPaving

Length of fencing real lengthFencing

Width of fencing real widthFencing

**Constant:** Pool Area real

63 ~ (9 \* 7) It could also have been an integer.

**Output:** Price that paving will cost real pricePaving Label Price that fencing will cost real priceFencing Label

Copyright reserved 14 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

**Pseudocode**

CalculatePrice

~ Enter input values

Display “Enter width of paving around the pool in meters:”

Enter widthPaving

Display “Enter width of garden around paving in meters:”

Enter widthGarden

Display “Enter cost of paving per square meter:”

Enter costPaving

Display “Enter cost of fencing per meter:”

Enter costFencing

~ Calculations for paving

totalLengthPaving = 9 + 2 \* widthPaving ~ Add width of paving to both sides of length of pool

totalWidthPaving = 7 + 2 \* widthPaving ~ Add width of paving to both sides of width of pool

totalArea = totalLengthPaving \* totalWidthPaving ~ total area of pool and paving

areaPaving = totalArea – areaPool ~ Deduct the area of the pool from the total area to get the ~ area of the paving only

pricePaving = areaPaving \* costPaving

~ Calculations for fencing

lengthFencing = totalLengthPaving + 2 \* widthGarden ~ Add width of garden to both sides of length of paving widthFencing = totalWidthPaving + 2 \* widthGarden ~ Add width of garden to both sides of width of paving priceFencing = costFencing \* (2 \* lengthFencing + 2 \* widthFencing) ~ Fencing all around garden

~ Therefore perimeter is needed

~ Display answers

display “Price for the paving = R”, pricePaving

display “Price for the fencing = R”, priceFencing

end

**Example 2**

**Problem statement**

Do the necessary planning and write an algorithm in pseudocode for the following:

Enter the following input fields:

• Student number.

• Student name. (E.g. initials and surname.)

• Marks obtained in a test for Principles of Computer Programming.

• The total for the test.

Convert this mark to a percentage and display the percentage on the screen in a meaningful format that will include the student name and student number. It must also include the status i.e.:

• (Fail) if the percentage is less than 50%.

• (Pass) if the percentage is from 50% to 74.9%; and

• (Pass with Distinction) if the percentage is 75% or above.

Many students did not perform well in this test and therefore the department decided to increase all final marks with 10%. Display the new percentage as well as the status of this new percentage. Display both answers in a user-friendly way. You may assume that all input values will be valid.

**IPO**

| I | P | O |
| --- | --- | --- |
| stName  stNumber  testMark  totalTest | Prompt to read input values  Enter input values  Calculate percentage and determine status1 Calculate new percentage and determine status2 Display answers | percentage  status1  newPercentage  status2 |

Copyright reserved 15 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

**Definitions and descriptions of the variables and controls to be used Description Type Name of variable Control**

**Input:** Student name (initials and surname) string stName Textbox Student number string stNumber Textbox Mark that student obtained for test integer testMark NumericUpDown Total for the test integer totalMark NumericUpDown

**Output:**Percentage mark that student obtained real percentage Label Status – Fail, Pass or Pass with Distinction string status1 Label New percentage after it has been increased by 10% real newPercentage Label Status after percentage has been increased by 10% string status2 Label

**Pseudocode**

CalculatePercentages

**~ Enter input values**

Display “Enter student initials and surname:”

Enter stName

Display “Enter student number:”

Enter stNumber

Display “Enter mark that student obtained for the test:”

Enter testMark

Display “Enter the total mark for the test:”

Enter totalMark

**~ Convert test mark to a percentage and add 10% to it, determine the status of each**

percentage = testMark \* 100 / totalMark

newPercentage = percentage \* 1.1

if percentage < 50 then

status1 = “ (Fail)”

else

if percentage < 75 then

status1 = “ (Pass)”

else

status1 = “ (Pass with distinction)”

endif

endif

if newPercentage < 50 then

status2 = “ (Fail)”

else

if newPercentage < 75 then

status2 = “ (Pass)”

else

status2 = “ (Pass with distinction)”

endif

endif

**~ Display answers Applicable labels to be used** display “Percentage for “, stName, “ (“, stNumber, “) = “, percentage, “%”, status1

display “After the additional 10% the mark is “, newPercentage, “%”, status2

end

Copyright reserved 16

**NATIONAL CERTIFICATE (VOCATIONAL) NQF LEVEL 3**

**REVIEWED ICASSPRACTICAL ASSESSMENT**

**TASK 2**

INSTRUCTIONS TO STUDENTS

**SUBJECT: PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3**

**SUBJECT CODE: 10041023**

**DATE OF IMPLEMENTATION: 01 JANUARY 2019** This document consists of 10 pages.

Copyright reserved Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

This is a practical application task that tests your ability to apply the knowledge and skills acquired in the vocational subject. The work required from you is an important and compulsory component of your studies.

**1. General information**

1.1 Study the task carefully.

1.2 Note the mark and time allocations per sub-task and for the task as a whole. 1.3 Note the submission date(s).

1.4 Study the resources required carefully.

1.5 Use the assessment tool as guidance on how your work will be marked. 1.6 You are required to provide proof that you submitted work done by yourself.

1.7 Save your work regularly. Keep a backup copy of this task. You may need to use this task for the ISAT.

**Preamble**

Before you start with practical assessment task 2, ensure that you understand and have sufficient practice in all the outcomes of Topics 2 and 3 of your curriculum, such as: • Developing VB.NET applications that incorporate different types of arrays, including working with parallel arrays.

• Defining and declaring structures.

• Using an array of structures as part of an application.

• Reading and writing to text files.

Remember that it is *your responsibility* to practice writing code and solving problems. In doing so you will become a skilled developer.

**2. Description of the task**

In practical assessment task 1, you developed an application that will allow users to make decisions and determine possible combinations of products that could be bought together. As part of your interface design, you captured data regarding the:

• customer, and the

• products on offer.

As part of practical assessment task 2, you are required to develop an application with an UI that will allow a user to perform certain functions.

First read the rest of the instructions and then decide on *one* of the following three requirements that you will implement as part your solution. You must choose *only one* of the following:

• Create a GUI application to manage the details of the customers; or

• Create a GUI application to manage the details of the stock that the store has; or • Create a GUI application to manage the various specials on offer.

Copyright reserved 2 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

For each of these you are required to store the data of at least 4 to a maximum of 6 different fields. For example, if you decide your application is to store data on the customers, you need to have fields such as customer number, customer details, address, email and phone number.

Your application will work and manage data stored in arrays. You must decide to use *one* of the following approaches:

• Implement a set of parallel arrays, where each array is used for a field; or • Implement an array with a structure, i.e. an array of structures.

Your application should read the data to populate your arrays from a text file. The text file content could be presented in any format such as fixed field widths, delimited fields, comma separated values, etc.

You must create a text file to use as the source of data for your array. There should be sufficient data in the text file to represent at least 15 records, i.e. array elements.

Read the data from the text file into your applicable array/s. Use an appropriate event for this purpose. The coding for this would require you to implement FileIO operations and various string methods.

Data operations are crucial for any data-driven application. Your application should include *one* the following functionalities:

• Must allow the user to browse or view the fields of each of the ‘records’; • Change two of the fields in an appropriate process, e.g. increase a price or change the stock quantity.

In addition, your application should include one of the following two functionalities: • Sorting and displaying the records based on a certain criterion; or

• Searching for an applicable record and displaying the data.

You should illustrate your ability to include coding functionality that illustrates your own creativity. This entails including at least 1 user-defined subroutine/function to facilitate some or other functionality. Finally, your application should save the data in the arrays back to a text file in an appropriate event.

The various task requirements are as follows:

• Design of the application UI.

• Definition of the applicable array/s.

• Definition and use of user defined function/subroutine.

• Populating the array with the data from the text file.

• One of the following two operations:

o Data operation 1 – browse the data in the array/s; or

o Data operation 2 – changing field values.

• One of the following two operations:

o Sorting the array/s based on a criterion; or

o Search functionality.

• Own creative code implementation.

• Writing the data back to the text file.

• Effective coding, proper comments and indentation applied.

Copyright reserved 3 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

As part of planning, include the following:

• The UI design with an applicable tool, as explained for practical assessment task 1. • The IPO and corresponding pseudocode for any two of the process operations of the application, e.g. sorting/searching or data operation 1, etc.

You should also ensure that you understand the application that you have developed and be able to demonstrate the application to your lecturer.

Create a well-formatted document with your source code, screenshots of your running application and explanations of the UI you have designed.

**3. Instructions for the task**

**3.1 Sub-task 1: Information gathering and program design** (Total: 15 marks)

This sub-task must be done in your own time of approximately 5 hours in preparation for the main task.

3.1.1 Gather information and design your UI in preparation for developing the program.

• Gather sample code of similar applications and examples of code where parallel arrays and structures are implemented. Another possible sample code includes:

o Text file operations, i.e. text file operations; and

o String manipulation methods and routines.

• Add your own notes regarding the meaning and the implementation of the example code.

Your lecturer should be able to provide you with detailed examples of code written by him/herself and you may find the following links useful:

• https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/programming/drives-directories files/file-access

• https://docs.microsoft.com/en-us/dotnet/visual-basic/developing-apps/programming/drives-directories files/how-to-read-from-comma-delimited-text-files

• https://www.tutorialspoint.com/vb.net/vb.net\_file\_handling.htm

• https://www.homeandlearn.co.uk/NET/nets8p2.html

• https://www.dougv.com/2007/07/working-with-a-simple-structure-array-in-vbnet/

• http://www.technologyuk.net/software-development/vbnet/arrays-and-structures.shtml

• https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/functions/string-functions • https://www.homeandlearn.co.uk/NET/nets7p1.html

3.1.2 Design a UI using an appropriate electronic tool. (See practical assessment task 1.) Name and explain the purpose of each of the components that you will implement as part of your application. Use the template supplied by your lecturer for practical assessment 1.

3.1.3 Develop applicable IPO charts and corresponding sets of pseudocode for *any two* of the process operations of the application, e.g. sorting/searching or data operation 1, etc.) Refer to the examples in Annexure A of practical assessment task 1 for this purpose.

3.1.4 Create the text file used to programmatically populate the array/s.

Copyright reserved 4 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

**3.2. Sub-task 2: Coding and debugging the program** (Total: 30 marks; 150 minutes)

This task must be done under supervision. Use the interfaces and the algorithms created in sub-task 1 to code the program.

3.2.1 Create a corresponding VB.NET project and form based on your UI design. Ensure that all components are properly named and that the solution is saved. (2)

3.2.2 Include the necessary declarations of the arrays (and structure) to be used. (2) 3.2.3 Code an applicable event to populate the array/s with the data of the text file. (5) 3.2.4 Write the code for a relevant user-defined subroutine/function. (3)

3.2.5 Write code to browse the data in the arrays, i.e. the user must be able to see the data as records in components; or

Write code to perform changes to the values of the data in the arrays, e.g. updating prices or removing stock, etc. (5)

3.2.6 Include the code for either the sort *or* search options. Ensure that you clearly comment the code. (5)

3.2.7 Own creative code implementation. (2)

3.2.8 Writing the data back to the text file. (5)

3.2.9 Code presentation/indentation/comments. (1)

**3.3 Sub-task 3: Demonstration, presentation and documentation** (Total: 5 marks)

This sub-task must be done under supervision following sub-task 2. The sub-task involves documenting the program by adding comments to the code, printing and submitting the software evidence. Follow the guidelines below to complete the sub-task.

3.3.1 Create a professionally formatted document that includes the documents of the previous stages; screenshots of your application in the VB.NET IDE as well as of your application running with different sample data. (3)

3.3.2 Present your application to your lecturer. For this task you need to be knowledgeable of the code you have written, as well as provide motivation regarding your design. (2)

3.3.3 Submit a soft copy of the program to your lecturer at the end of the demonstration.

**4. Time and mark allocation**

| **Sub-task** | **Activity** | **Time**  **allocation** | **Mark**  **allocation** | **Assessment tools** |
| --- | --- | --- | --- | --- |
| 1 | Information gathering and program design. (Student’s own time) |  | 15 | Checklist |
| 2 | Developing, coding and debugging. (Supervised) | 150 minutes | 25 | Checklist |
| 3 | Documentation, presentation and  submission. (Supervised) | 30 minutes | 5 | Checklist |
| **TOTAL** | | **180 minutes** | **50** |  |

Copyright reserved 5 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

**5. Assessment tool**

Use the checklist below as guidance on how your work will be marked. The mark allocation is a guide to assess each sub-task. Your response will be assessed in terms of the quality, appropriateness and level of effort/analysis you have put in.

| **STUDENT’S CHECKLIST: PRACTICAL ASSESSMENT TASK 2** | | |
| --- | --- | --- |
| **Task description**  GUI application based on practical assessment task 1, incorporating data operations. | **SECTION MARK** | **TICKLIST** |
| **Sub-task 1: Information gathering and program design** | **[15]** |  |
| *3.1.1 Information gathering*  *Code examples and snippets with own notes and explanations – criteria:* • The presentation contained at least 5 different examples of applicable code related to the scenario, e.g.  o Text file operations.  o Arrays.  o Structures.  o Arrays or structures.  o Parallel arrays.  o Sorting.  o Searching.  o Array operations.  • All the examples contained detailed notes/explanations made by the student and were neatly grouped and labelled.  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| *3.1.2 GUI design and composition – criteria*  • A detailed wireframe created with a software tool was presented. • The wireframe indicated the various components and were grouped in the given format.  • The wireframe was presented neatly with the controls aligned and was suitable for its purpose.  • The wireframe contained descriptions for each component and a suitable name. • All controls selected and included were motivated.  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |
| 3.1.3 *IPO charts and pseudocode – criteria*  • *Two* IPO charts were created with the input variables, basic algorithm and output variables identified.  • The algorithms were correct in relation to the variables and the task at hand. • Each IPO had a corresponding description of the input, possible intermediate, possible constants and output variables identified.  • For each IPO all the variables were identified and a suitable datatype and name were given. Where applicable, a suitable component was indicated.  • IPO Operation 1 –  o The pseudocode was presented in a readable way with clear indications of the logical flows and constructs required.  o The appropriate input, processing and output operations (or in the case of a subroutine, the parameters being passed) were identified and used.  o The pseudocode related to the algorithm and was logically correct. o The code was indented and presented neatly with comments where applicable. |  |  |

Copyright reserved 6 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| • IPO Operation 2 –  o The pseudocode was presented in a readable way with clear indications of the logical flows and constructs required.  o The appropriate input, processing and output operations (or in the case of a subroutine, the parameters being passed) were identified and used.  o The pseudocode related to the algorithm and was logically correct. o The code was indented and presented neatly with comments where applicable. • A well-formatted document including all the required details as stipulated was presented.  • The content of the logical planning document was neatly presented and there was evidence of pride and professionalism in the presentation thereof.  (8 = all the criteria are met; 5 = one criterion is not met; 3 = two criterions are not met; 1 = three criterions are not met; 0 = more than three criterions are not met.) | 8 |  |
| *3.1.4 Design and content of the text file to be used as a data source to populate the array/s or array structures – criteria*  • The textfile contains sufficient data for at least 15 records  • Each line of text presents a suitable record that has at least 4 fields • An appropriate mechanism is implemented that will enable a programmer to extract the fields  o Fixed field with approach used; or  o Appropriate delimiter used; or  o Combination of fixed field widths and delimiters  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| **Sub-task 2: Coding and debugging the program** | **[30]** |  |
| *3.2.1 VB.NET project and UI design – criteria:*  • A VB.NET forms-based application was created which was appropriately named and saved.  • All files were named correctly, and the project had a relevant name. • The GUI design corresponded with the wireframe design.  • All components implemented were appropriately named.  • All components implemented were suitable for their applicable tasks. (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| *3.2.2 Array and structure – criteria:*  • At *least* 4 parallel arrays were declared.  • The arrays were appropriately named.  • The size of the arrays was all the same.  • The arrays related to the “fields” as presented in the text file.  • The arrays were appropriately initialised.  OR  *The student implemented parallel arrays - criteria*  • A structure was defined that included various fields related to the data of the text file records.  • The fields were correctly named with suitable data types.  • The structure contained at least 4 fields.  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |

Copyright reserved 7 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| *3.2.3 Code an applicable event to populate the array/s with the data of the text file – criteria*  • An applicable Streamreader or other reader object for file access was defined. • The external file was opened and referenced appropriately.  • An applicable mechanism was built with code to manage whether the file could be accessed or not and if not, appropriate action was to be taken.  • The text file reference object was used correctly within the code constructs used to process the file.  • An applicable sentinel controlled looping structure was utilised to read through the text file line by line.  • There was evidence that the student understood the concepts of string and index manipulation in relation to the code construct developed.  • A line of text from the text file was assigned to an appropriate variable. • Various string manipulation methods were applied to extract the various fields from the line of text.  • An index for array referencing purposes was declared.  • The various fields were assigned to the applicable array (in the case of a structure the array of structure fields) or elements.  • The index was appropriately managed and incremented.  • Any final operations regarding the processing were implemented. (5 = all the criteria are met; 3 = one criterion is not met; 2 = two criterions are not met; 1 = three criterions are not met; 0 = more than three criterions are not met.) | 5 |  |
| *3.2.4 Relevant user defined subroutine/function – criteria:*  • The subroutine or function included had a particular purpose as part of the application. • The subroutine had all the necessary parameters and will function according to scope and the logical constructs needed.  • The required intermediate calculations were done correctly.  • Where applicable, the parameters were declared and used correctly. • Where applicable, the correct value was returned, or reference parameters used. • The function was called appropriately within the larger application and the code  presented evidence of knowledgeable use and implementation of user-defined functions or routines.  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |
| *3.2.5 Browse data in the arrays OR perform changes in the data of the arrays – criteria* • The student code implemented worked perfectly and showed evidence that the student fully understood the task, e.g. browsing or updating of data elements. • The logic applied by the student achieved the required task.  • The changes to the elements of the array were applied correctly.  • The index was used and updated appropriately.  • The code included a mechanism to show the changes to the user. • The code was neatly presented, indented and commented.  (5 = all the criteria are met; 3 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 5 |  |

Copyright reserved 8 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| *3.2.6 Search OR Sort – criteria*  • The student code implemented worked perfectly and showed evidence that the student fully understood the task, e.g. sorting or searching elements.  • The logic applied by the student achieved the required task.  • The comparison of the elements of the array was applied correctly to perform the required task.  • The index was used and updated appropriately.  • The code included a mechanism to show the changes to the user. • The code was neatly presented, indented and commented.  (5 = all the criteria are met; 3 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 5 |  |
| *3.2.7 Own creative code implementation – criteria*  • Student was able to identify and explain two areas in the application code where additional features (or creative code) and enhancements were added. • Each area should contain at least 2-3 logical code constructs.  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| *3.2.8 Writing the data back to a text file – criteria:*  • An applicable StreamWriter or other reader object for file access was defined. • The file external was created and referenced appropriately.  • An applicable sentinel controlled looping structure was utilised to read through the array/s element by element.  • A line of text was created to be written to the file.  • The line of text was correctly written to the file.  • After the operation was completed, the text file was saved and when opened and viewed, the data was updated as intended.  (5 = all the criteria are met; 3 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 5 |  |
| *3.2.9 Code presentation/indentation/comments – criteria:*  • Local variables were properly defined and included.  • The code was indented appropriately.  • Sufficient comments were included.  • The presentation of the code was professionally done, e.g. naming of components. • Use of appropriate constructs and methods are applied  (1 = all the criteria are met; 0 = one criterion is not met.) | 1 |  |
| **Sub-task 4: Demonstration, presentation and documentation** | **[5]** |  |
| *3.4.1 Application document – criteria*  • The created document was based on a template and included all the various elements of the previous sub-tasks.  • Uniform fonts were applied, and the document had a professional appeal. • The document contained a cover page with the appropriate details. • An index was included.  • Applicable running screenshots with descriptions were included as part of the user documentation.  • There was evidence that the student took pride in his/her work and the document could be used for reference purposes by future students.  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |
| *3.4.2 Demonstration – criteria*  • The student comfortably demonstrated the application and was knowledgeable on all the code constructs implemented by answering all questions correctly. • The application worked perfectly as intended.  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| **TOTAL MARKS** | **50** |  |

Copyright reserved 9 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

| **6. RECORD OF PERFORMANCE IN ICASS PRACTICAL ASSESSMENT TASK 2** | | | |
| --- | --- | --- | --- |
| **College** |  | | |
| **Campus** |  | | |
| **Student’s Surname and First Name/s** |  | | |
| **Student’s ID Number** |  | | |
| **Lecturer’s Surname and Initials** |  | | |
| **Date of conclusion of assessment** |  | | |
| **ASSESSMENT GRID** | | | |
| **SUB-TASKS** | | **MARK ALLOCATION** | **STUDENT’S MARK** |
| Sub-task 1: Information gathering and program design | | 15 |  |
| Sub-task 2: Coding and debugging the program | | 30 |  |
| Sub-task 3: Demonstration, presentation and documentation | | 5 |  |
| **Total** | | **50** |  |
| **Converted to %** | | **100%** |  |

| **COMPETENCE LEVEL INDICATORS** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Outstanding** | **Highly**  **competent** | | **Competent** | **Not yet competent** | **Not achieved** |
| **80-100%** | **70-79%** | | **50-69%** | **40-49%** | **0-39%** |
| **5** | **4** | | **3** | **2** | **1** |
| **Student’s Competence Level** | |  | | | |
| **Student’s Signature** | |  | | | |
| **Lecturer’s Signature** | |  | | | |
| **Date** | |  | | | |
| **Moderator’s Signature** | |  | | | |
| **Feedback to student** | | | | | |

Copyright reserved 10

**NATIONAL CERTIFICATE (VOCATIONAL) NQF LEVEL 3**

**REVIEWED INTEGRATED**

**SUMMATIVE ASSESSMENT TASK** INSTRUCTIONS TO STUDENTS

**SUBJECT: PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3**

**SUBJECT CODE: 10041023**

**DATE OF IMPLEMENTATION: 01 JANUARY 2019** This document consists of 13 pages.

Copyright reserved Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

This is a practical application task that tests your ability to apply the knowledge and skills acquired in the vocational subject. The work required from you is an important and compulsory component of your studies.

**1. General information**

1.1 Study the task carefully.

1.2 Note the mark and time allocations per sub-task and for the task as a whole. 1.3 Note the submission date(s).

1.4 Study the resources required carefully.

1.5 Use the assessment tool as guidance on how your work will be marked. 1.6 You are required to provide proof that you submitted work done by yourself. 1.7 Save your work at regular intervals. Keep a backup copy of your work.

**Preamble**

Before you start with the ISAT, ensure that you understand and have sufficient practice in all the outcomes of Topics 2, 3, 4 and 5 in your curriculum, such as:

• Developing VB.NET applications that incorporate different types of arrays (including working with parallel arrays.)

• Defining and declaring structures.

• Using an array of structures as part of an application.

• Reading and writing to text files.

• Applying defensive programming techniques and methods.

• Applying user input validation.

In addition, you should thoroughly practice using MS Access by creating your own tables, setting up and defining keys and relationships, developing and designing forms, queries and reports.

Remember that it is *your responsibility* to practice writing code and solving problems. In doing so you will become a skilled developer.

**2. Description of the task**

In the first and second practical assessment tasks, you investigated the use of VB.NET to create a GUI-based application that allows users to perform certain tasks.

The first practical assessment task focussed on the development of an application that allows users to select products and accessories. Based on the products and accessories or combinations of other products, certain discounts could have applied.

In your second practical assessment task, the focus was on allowing the user to browse the various products (i.e. stock); the various specials or customers.

Copyright reserved 2 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

For the ISAT, you are required to develop two separate applications, namely: • VB.NET application.

• Ms Access application.

The two applications are related. Remember how you were tasked to create an output text file as part of your practical assessment task 2 application? Often text files become the source of data to populate tables as part of a database-driven application.

**Application 1: VB.NET**

For this task, you are required to enhance and extend your practical assessment task 2 project. It is a perfect opportunity to learn more and become a better developer.

In practical assessment task 2, you were given the choice to select and include various functionalities as part of your solution (refer to the task description). These were: • One of the following two operations:

o Data operation 1 – browse the data in the array/s; or

o Data operation 2 – changing field values.

• One of the following two operations:

o Sorting the array/s based on a criterion; or

o Search functionality.

The first objective is to include the *two* functionalities that you did not originally select to do as part of your application. Secondly you are required to include and apply two different data validation techniques on two different elements of your application. Examples of various validation techniques are:

• Range or constraint checks.

• Verifying masks and input fields.

• Applying checksums or related algorithms.

• Data type checks.

• Consistency checks.

• Etcetera.

One of the validation techniques should be done by implementing a user-defined function that returns a Boolean value.

Thirdly, you are required to apply *one* defensive programming technique to an appropriate event of your application. Examples of defensive programming techniques are: • Catering for unexpected data or occurrences.

• Highlighting potential issues.

• Adding guards and exception handling.

**Application 2: Ms Access**

For this task, you must identify any two related entities using the data as part of the output text file you have created for practical assessment task 2. Refer to Annexure B for an example in which three entities are identified.

You must design a basic ERD with *two* entities that also include attributes (i.e. fields). You may add applicable attributes/fields to the entities you design.

Copyright reserved 3 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

From the ERD and using Ms Access, you must create an appropriate project with *two* related tables that will provide the user with the opportunity to browse and edit data, run queries and reports. Your Ms Access application will be related to your VB.NET application as it is based on the same data.

You are allowed to be creative in including any other functionality that adds value to the program within the allocated timeframe.

**3. Instructions for the task**

3.1 **Sub-task 1: Information gathering, program and database design** (Total: 15 marks)

You are required to do the following in preparation for the ISAT in your own time of approximately 5 hours.

3.1.1 Create an IPO chart and a corresponding set of pseudocode for one of the tasks you did not include as part of practical assessment task 2. (Refer to the description of the task.) (4)

3.1.2 Adapt your GUI design to incorporate the applicable controls for the newly added features. Include these in your wireframe. (2)

3.1.3 Research the difference between validation and defensive programming and how it could be applied in a VB.NET application. Write a one-page summary on each concept in your own words. You may include code snippets. Reference all sources appropriately. Include a cover page, 2 content pages and 1 bibliography page. You need to include at least six

different sources. (3)

3.1.4 Create an IPO chart and a corresponding set of pseudocode for a user-defined function that you will use as part of the validation operations of your application. Your function returns a Boolean value. (3)

3.1.5 Using software such as Draw.io or Ms Visio, create an ERD based on your output text file as described in the task description. Your ERD should comprise of:

• At least *two* entities.

• At least *one* relationship.

• Appropriate fields.

• An indication of all the keys. (3)

3.2 **Sub-task 2: Enhancement of functionality** (Total: 35 marks; 60 minutes) This task will be done under supervision.

You are required to use your practical assessment task 2 application and add the modifications below.

3.2.1 Add the missing data operation functionality (VB.NET code) (typically as part of an event) to the application that you *did not* include as part of the task. (Browse the data or change field values.) (10)

3.2.2 Add the other missing operation functionality to your application. (Sort or search.) (10)

Copyright reserved 4 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

3.2.3 Write the code that will incorporate a data validation technique on one of the data elements of your application. Include the user-defined function you have designed in 3.1.4. (5)

3.2.4 Write applicable code for a second data validation operation on one of the other data elements of your application. (5)

3.2.5 Include code in your application that illustrates that you understand the concept of defensive programming. (5)

3.3 **Sub-task 3: Ms Access Application** (Total: 40 marks; 120 minutes) This task will be done under supervision.

You are now required to create a Ms Access application.

3.3.1 Create, name and save your Ms Access application. (1)

3.3.2 Create two tables that correspond to the ERD that you have designed. One of your tables should contain at least six fields and the other at least three. Include appropriately named fields and keys, also use appropriate data types for each field. (6)

3.3.3 Add at least fifteen records to one of your table and at least five records to another table. The data of one of your tables should closely correspond to the text file you have used. (Tip: you could import the data to one of your tables.) (4)

3.3.4 Include a meaningful relationship with referential integrity between your two tables. (2) 3.3.5 Create *two* forms for easy input of data and ensure data validation during input. For one of the forms you should make use of combo boxes and buttons to assist with the data capturing. You need to include at least *three* design changes to one of your forms. Ensure that all labels are clear and meaningful. (12)

Extra marks will be awarded for the inclusion of a sub-form in a main form or switchboard which links multiple forms.

3.3.6 Create at least three meaningful queries, which include advanced criteria and expressions/calculations.

• One of your queries should include a suitable join.

• One of your queries should contain a parameter.

• One of your queries must contain a calculation (calculated field).

• One of your queries need to include a sort.

• One of your queries should implement a relational operator as part of a condition. (8) 3.3.7 Create at least two meaningful reports, grouped according to specific fields that include calculations on fields per group and/or per report.

• Ensure that one of your reports is grouped on more than one field level. • Add a group total or value to one of your reports.

• Ensure that your column headings are well-formatted.

• Include appropriate report headings for each of your reports.

• Include the image or company logo as part of one of your report headings. (7)

Copyright reserved 5 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

3.4 **Sub-task 4: Program documentation and presentation** (Total: 10 marks; 60 minutes) This task will be done under supervision.

You are now required to document and present your:

• Enhanced practical assessment task 2 VB.Net application; and

• Ms Access application.

*Enhanced VB.Net application*

3.4.1 Create a professionally formatted document that includes the documents of the previous stages; screenshots of your application in the VB.NET IDE as well as of your application running with different sample data. (3)

3.4.2 Present your application to your lecturer. For this task, you need to be knowledgeable about the code you have written, as well as provide motivation regarding your design. (2)

3.4.3 Submit a soft copy of the program to your lecturer at the end of the demonstration. *Ms Access application*

3.4.4 Create a professionally formatted document that includes screenshots of the various elements of your Ms Access application. Table designs, relationships, data, forms, queries and reports. Add suitable descriptions to each. (3)

3.4.5 Present your application to your lecturer. For this task, you need to be knowledgeable about your database and the corresponding elements, as well as provide motivation regarding your design. (2)

3.4.6 Submit a soft copy of your Ms Access file to your lecturer at the end of the demonstration.**4. Time and mark allocation**

| **Sub-task** | **Activity** | **Time**  **allocation** | **Mark**  **allocation** | **Assessment**  **tools** |
| --- | --- | --- | --- | --- |
| 1 | Information gathering, program and database design |  | 15 | Checklist |
| 2 | Enhancing functionality | 60 minutes | 35 | Checklist |
| 3 | Ms Access application | 120 minutes | 40 | Checklist |
| 4 | Program documentation and presentation | 60 minutes | 10 | Checklist |
| **TOTAL** | | **240 minutes** | **100** |  |

**5. Assessment tool**

Use the checklist on page 7 as guidance on how your work will be marked. The mark allocation is a guide to assess each sub-task. Your response will be assessed in terms of the quality, appropriateness and level of effort/analysis you have put in.

Copyright reserved 6 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

| **STUDENT’S CHECKLIST: INTEGRATED SUMMATIVE ASSESMENT TASK** | | |
| --- | --- | --- |
| **Task description**  GUI application based on practical assessment task 1, incorporating data operations. | **SECTION MARK** | **TICKLIST** |
| **Sub-task 1: Information gathering, program and database design** | **[15]** |  |
| *3.1.1 IPO chart and pseudocode – additional operation* ***– c****riteria:*  • The IPO chart was created with the input variables, basic algorithm, and output variables identified.  • The algorithm was correct in relation to the variables and the task at hand. • The IPO had a corresponding description of the input, possible intermediate, possible constants and output variables identified.  • For each of the variables a suitable datatype and name was given. Where applicable, a suitable component was also indicated.  • The pseudocode was presented in a readable way with clear indications of the logical flows and constructs required.  • The appropriate input, processing and output operations (or in the case of a subroutine, the parameters being passed) were identified and used.  • The pseudocode related to the algorithm and was logically correct. • The code was indented and presented neatly with comments where applicable. (4 = all the criteria are met; 2 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 4 |  |
| *3.1.2 GUI design and composition – criteria:*  • A detailed wireframe created with a software tool was presented. The wireframe indicated the various components and was grouped in the given format. • The wireframe was presented neatly with the controls aligned and was suitable for its purpose.  • The wireframe contained descriptions for each component and a suitable name. (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| *3.1.3 Summary of the concepts of validation and defensive programming related to VB.NET application development – criteria:*  • The document comprised of a cover page with the student’s details and a title. • Two pages were included and the content presented clearly discussed and differentiated between the two concepts.  • Sample code to illustrate the concepts was included.  • From the composition of the form it was evident that the student had not plagiarised and that the work was done by him/herself. The document was neatly presented with uniform fonts and the document exhibited a sense of pride.  • A bibliography was included with at least six different sources.  (4 = all the criteria are met; 2 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 4 |  |
| *3.1.4 IPO chart – user-defined function – criteria:*  • The IPO chart was created with the input variables, basic algorithm, and output variables identified.  • The algorithm was correct in relation to the variables and the task at hand. • The pseudocode was presented in a readable way with clear indications of the logical flows and constructs required.  • The appropriate input, processing and output operations (or in the case of a subroutine, the parameters being passed) were identified and used.  • The pseudocode related to the algorithm and was logically correct. • The pseudocode was indented and presented neatly with comments where applicable. (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |

Copyright reserved 7 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| *3.1.5 ERD design from text file – criteria:*  • The ERD was designed in an appropriate electronic tool. (Not MS Access.) • Two *entities* were included and correctly named (in the singular).  • Each entity had an appropriate PK and FKs.  • Each entity had applicable fields/attributes.  • The relationship between the two entities was correctly indicated and named (with a verb).  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 3 |  |
| **Sub-task 2: Enhancement of functionality** | **[35]** |  |
| *3.2.1 Including missing data operation functionality to practical assessment task 2 project (Browse the data or change field values) – criteria:*  • The student code implemented worked perfectly and showed evidence that the student fully understood the task at hand, e.g. browsing or updating of data elements. • The logic applied by the student achieved the required task.  • The changes to the elements of the array were applied correctly.  • The index was used and updated appropriately.  • The code included a mechanism to show the changes to the user. • The code was neatly presented, indented and commented.  (10 = all the criteria are met; 7 = one criterion is not met; 5 = two criterions are not met; 2 = three criterions are not met; 0 = more than two criterions are not met.) | 10 |  |
| *3.2.2 Adding other missing operation functionality (Sort or Search)* ***–*** *criteria:* • The student code implemented worked perfectly and showed evidence that the student fully understood the task at hand, e.g. browsing or updating of data elements. • The logic applied by the student achieved the required task.  • The changes to the elements of the array were applied correctly.  • The index was used and updated appropriately.  • The code included a mechanism to show the changes to the user. • The code was neatly presented, indented and commented.  (10 = all the criteria are met; 7 = one criterion is not met; 5 = two criterions are not met; 2 = three criterions are not met; 0 = more than two criterions are not met.) | 10 |  |
| *3.2.3 User defined Boolean function for data validation – criteria:*  • The function included did perform a data validation operation.  • The function had all the necessary parameters and will function according to scope and the logical constructs needed.  • The required intermediate calculations were done correctly.  • Where applicable, the parameters were declared and used correctly. • Where applicable, the correct value was returned, or reference parameters used. • The function was called appropriately within the larger application. • The student code presented evidence of the knowledgeable use and implementation of user defined functions or routines.  (5 = all the criteria are met; 3 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 5 |  |
| *3.2.4 2nd Validation implementation – criteria:*  • The code included as part of an event did perform a data validation operation. • The logic applied was evident in the code and was sufficient for the applicable field chosen.  • The validation approach used in the code, worked and illustrated that the student understood the concept.  (5 = all the criteria are met; 3 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 5 |  |

Copyright reserved 8 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| *3.2.5 Defensive programming technique applied – criteria:*  • The code included for this purpose worked and was applicable.  • The defensive programming operation was clearly highlighted, and communication was presented to the user.  • The defensive approach used in the code, worked and illustrated that the student understood the concept.  (5 = all the criteria are met; 3 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 5 |  |
| **Sub-task 3: Ms Access application** | **[40]** |  |
| *3.3.1 Creating, naming and saving the Ms Access application*  • The application had an appropriate name and was saved. (1 mark) | 1 |  |
| *3.3.2 Creating MS Access tables based on the ERD – Tables – criteria:* • Two tables were created and appropriately named.  • One table contained at least six fields and the other three.  • All the fields had meaningful names and corresponded to the ERD. • Data types were appropriate and corresponded to the content of the fields. • Field properties were appropriate and corresponded to the content of the fields. • All key fields were indicated.  (6 = all the criteria are met; 4 = one criterion is not met; 3 = two criterions are not met; 1 = three criterions are not met; 0 = more than two criterions are not met.) | 6 |  |
| *3.3.3 Adding records to the tables – Table records – criteria:*  • At least fifteen related records were included in one of the tables. • The data of one table closely corresponded to the data of the text file. • At least five different records were included in the related table.  • Key fields were appropriate and contained valid data.  (4 = all the criteria are met; 2 = one criterion is not met; 1 = two criterions are not met; 0 = more than two criterions are not met.) | 4 |  |
| *3.3.4 Relationship created between the two tables – criteria:*  • A relationship was established between the two tables.  • Referential integrity was included.  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| *3.3.5 Input form creation – criteria (check each applicable):*  • Two forms were created.  • The forms were appropriately named.  • The forms will facilitate data input.  • There was evidence that the student applied one unique change to the form. • There was evidence of a second unique change.  • There was evidence of a third unique change.  • A combo box or applicable control to assist with data capturing was added. • At least *one* additional button was added to one of the forms.  • The fields included were relevant.  • All the labels were clear.  • The changes to the form were meaningful.  • It was evident that the student understood the concept and purpose of forms in Ms Access.  (1 mark per criteria = 12) | 12 |  |
| **Bonus marks**  *Input form creation – bonus mark criteria:*  • A dashboard was included as part of the application.  • A sub-form was included as part of one of the forms.  • The student understood the various enhancements.  (1 mark per criteria = 3) |  |  |

Copyright reserved 9 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| *3.3.6 Inclusion and creation of queries – criteria (check each applicable):* • Three queries are created and named appropriately  • All three queries have a specific purpose and present answers to information needs • One query contains a join with fields of more than one table  • One of the queries contain a parameter  • An expression or calculation is used on one query  • One of the queries include a sort  • One of the queries contain a condition with a relational operator  • The student is able to answer questions based on the SQL view of the queries he/she created  (1 mark per criteria = 8) | 8 |  |
| *3.3.7 Inclusion and creation of reports – criteria (check each applicable)* • At least two meaningful reports were included and appropriately named. • Each report contributed to the solution.  • One of the reports was grouped on one or more level.  • The grouped report contained a total.  • One of the reports contained the value of a calculated field.  • The column headings of the reports were well-formatted.  • One of the report headings included an image.  (1 mark per criteria = 7) | 7 |  |
| **Sub-task 4: Program documentation and presentation** | **[10]** |  |
| **Enhanced practical assessment task 2 VB.Net application** | 3 |  |
| *3.4.1 Document creation – criteria:*  • The created document was based on a template and included all the various elements of the previous sub-tasks.  • Uniform fonts were applied and the document had a professional appeal. • The document contained a cover page with the appropriate details. • An index was included.  • Applicable running screenshots with descriptions were included as part of the user documentation.  • There was evidence that the student took pride in his/her work and the document could be used for reference purposes by future students.  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) |
| *3.4.2 Demonstration – criteria:*  • The student comfortably demonstrated the application and was knowledgeable on all the code constructs implemented by answering all questions correctly. • The application worked perfectly as intended.  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| **Ms Access application** | 3 |  |
| *3.4.3 Document creation – criteria:*  • The created document was based on a template and included all the various elements of the previous sub-tasks.  • Uniform fonts were applied and the document had a professional appeal. • The document contained a cover page with the appropriate details. • An index was included.  • Applicable running screenshots with descriptions were included as part of the user documentation. These included table designs, relationships, data, forms, queries and reports.  • There was evidence that the student took pride in his/her work and the document could be used for reference purposes by future students.  (3 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) |

Copyright reserved 10 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

|  | **SECTION MARK** | **TICKLIST** |
| --- | --- | --- |
| *3.4.4 Demonstration – criteria:*  • The student comfortably demonstrated the application and was knowledgeable on all the elements such as tables, forms, keys, queries, reports, working in design view, etc.  • The student was able to answer all questions correctly.  • The application worked perfectly as intended.  (2 = all the criteria are met; 1 = one criterion is not met; 0 = more than one criterion is not met.) | 2 |  |
| **TOTAL** | **100** |  |

Copyright reserved 11 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

| **6. RECORD OF PERFORMANCE IN INTEGRATED SUMMATIVE ASSESSMENT TASK** | | | |
| --- | --- | --- | --- |
| **College** |  | | |
| **Campus** |  | | |
| **Student’s Surname and First Name/s** |  | | |
| **Student’s ID Number** |  | | |
| **Lecturer’s Surname and Initials** |  | | |
| **Date of conclusion of assessment** |  | | |
| **ASSESSMENT GRID** | | | |
| **SUB-TASKS** | | **MARK ALLOCATION** | **STUDENT’S MARK** |
| Sub-task 1: Information gathering, program and database design. | | 15 |  |
| Sub-task 2: Enhancement of functionality | | 35 |  |
| Sub-task 3: Ms Access application | | 40 |  |
| Sub-task 4: Program documentation and presentation | | 10 |  |
| **Converted to %** | | **100%** |  |

| **COMPETENCE LEVEL INDICATORS** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Outstanding** | **Highly**  **competent** | | **Competent** | **Not yet competent** | **Not achieved** |
| **80-100%** | **70-79%** | | **50-69%** | **40-49%** | **0-39%** |
| **5** | **4** | | **3** | **2** | **1** |
| **Student’s Competence Level** | |  | | | |
| **Student’s Signature** | |  | | | |
| **Lecturer’s Signature** | |  | | | |
| **Date** | |  | | | |
| **Moderator’s Signature** | |  | | | |
| **Feedback to student** | | | | | |

Copyright reserved 12 Please turn over

NC (V) PRINCIPLES OF COMPUTER PROGRAMMING LEVEL 3 2019

**Annexure B - Example of identifying entities and attributes from a text file as a source of analysis**

The following tab-delimited text file contains data from different athletes:

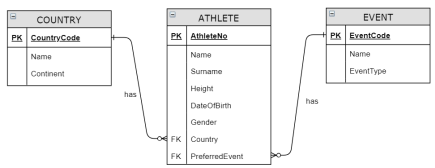


If we analyse the data, we could identify three possible entities, such as the:

• Athlete.

• Country of the athlete.

• Sport (or event) that the athlete competes in.

It is then possible for us to create the following simple ERD indicating the relationships between the entities: If we examine the ERD, we could deduce the following basic business rules:

• One *country* could have many *athletes*. Each *athlete* has a preferred *event* and each *event* has various *athletes* that could compete.

• For the fields (attributes) in the *country* table (entity) and the *event* table the designer thereof included related fields that are appropriate. The fields of the *athlete* table correspond to the fields as in the text file.

Copyright reserved 13