



## FACULTY OF SCIENCE

### ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

<b>MODULE</b>	<b>IFM01A1 &amp; IFM1A10</b> INFORMATICS 1A: INTRODUCTION TO ALGORITHM DEVELOPMENT (VB)
<b>CAMPUS</b>	APK
<b>EXAMINATION</b>	JULY 2019 – PAPER C (SSA)

**DATE** 2019-07-16 **TIME** 15:00–18:00

**ASSESSORS** K LEBEA  
WS LEUNG

**INTERNAL MODERATOR** HJC VAN DER WESTHUIZEN

**DURATION** 3 Hours **MARKS** 100

---

**PLEASE READ THROUGH THESE INSTRUCTIONS CAREFULLY AND ADHERE TO THEM**

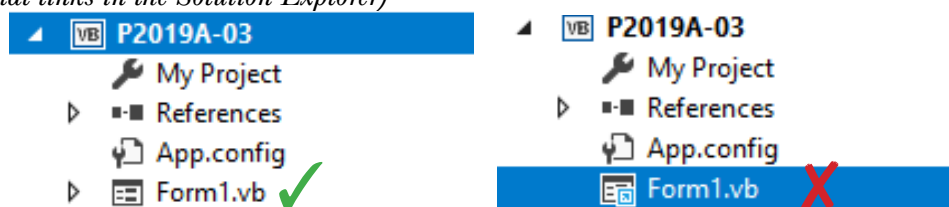
***The Academy takes NO responsibility for your ignorance (willful or otherwise) and will NOT grant additional opportunities to students who fail to submit their work correctly***

- ☞ This question paper comprises 4 (four) pages (including this cover page)
- ☞ The use of calculators are NOT permitted during this examination
- ☞ The first 20 (twenty) minutes are reserved for DESIGN (must be done in answer book provided)
- ☞ Answer ALL questions
- ☞ No further marks will be awarded for Correct Execution from the point your submitted program terminates unexpectedly
- ☞ Follow the project naming convention indicated on your number card.
- ☞ SAVE REGULARLY - no additional time will be granted due to power failures
- ☞ Save ONLY to the T:\ directory
- ☞ Do NOT save to other locations outside of the default project folder - use ONLY the Save All button
- ☞ Read through page 1 for submission guidelines when you are ready to submit. Be sure to complete your details
- ☞ An invigilator must be alerted to verify and take in your submission

**DO NOT TURN OVER THIS QUESTION PAPER UNTIL YOU  
HAVE BEEN GIVEN INSTRUCTION TO COMMENCE**

**When you are ready to submit your practical...**

- On your exam booklet, **CLEARLY** indicate your details, **INCLUDING** your Computer Number
- Preparing your submission
  - Zip up the folder containing your project - ensure that this zip contains your entire project
- Uploading to Eve
  - Visit <http://eve.uj.ac.za>
  - Select File Storage from the Menu
  - Select the Session Informatics 1A Examination
  - Enter your username (your student number)
  - Enter the password (found on the back of your number card placed at your computer)
  - Upload the Zip file you created in Step b)
- Verifying the Upload
  - Download the file you uploaded to another location
  - Unzip it to verify that the entire project is there (*make sure that your project does not show any external links in the Solution Explorer*)



- Preparing the Backup to CD (use the CD provided to you)
  - Burn **EVERYTHING** on the T:\ drive to the CD
  - Use your *Student Number* for the **Disc title** and Select **With a CD/DVD player**
  - If your CD fails, alert an assistant to backup your project to USB instead
- Agreement and Submission (*to be completed when you submit*)
  - Complete your details - by signing below, you acknowledge that you have done the above steps to ensure that your submission is correct.

**Checklist (check all boxes):**

- ☐ I have followed the naming convention as indicated on my password card
- ☐ A zip of the full and final project has been uploaded to Eve
- ☐ A backup of the full and final project has been made, either to CD or USB

*I further acknowledge that:*

- ☐ I am aware that a non-compiling submission is capped at 40%
- ☐ Code I comment out will not be marked
- ☐ I take full responsibility for ensuring that my submission is correct and the version I intend to submit for assessment

Student #										PC #			
ID/Passport #													
Signature													
Backup Used	CD						USB						

Hand everything in to the invigilators (*this question paper, the assessment script, CD, number card*).



Business has been booming for TT Painting and twins Tshwanelo and Thabang have branched out operations to produce their own paint range. They've approached you to design and develop a Visual Basic application that will help them to see which TT Paint is the least popular with customers.

0.1	Design ( <i>Input, Output, Events &amp; Actions, Variables, Record structures, Interface, Test Data, Algorithms</i> )	5
0.2	Implementation of Interface	2
0.3	Option Statements	1
0.4	Effective use of subroutines / functions	1
0.5	Commenting	1

Here are the requirements that the twins have specified:

- 1) The application must be able to store the following data for each Paint Product:

Field	Example Data
a) ID	"AZB"
b) Cost to Produce per Litre	7.19
c) Data on each Offering:	
i) Size (in litres)	2.5
ii) Selling price per Unit	200.00
iii) Units Sold per Week	15; 22; 14; ...; 18
iv) Total Profit	See Question 4
d) Least Profitable Offering	See Question 5
e) Paint Rating	See Question 7

1.1	Definition of Offering Type record structure	4
1.2	Definition of Paint Product record structure	5

- 2) The application must be able to handle any number of Paint Products, with each Product being offered in different sizes. Each size offered is monitored for sales across a number of weeks. These numbers are specified by the user at the beginning. You may assume that each Paint Product is offered for the same number of sizes and that each size is monitored for the same number of weeks.

2.1	Input of number of Paint Products, Offerings, and Weeks	2
2.2	Set up of dynamic arrays	5
2.3	Set up of UJGrid control	2
2.4	Labelling of UJGrid control headings ( <i>Paint Products as Rows, and Offerings as Columns</i> )	3

- 3) The user must be able to input the necessary data into the program.

3.1	Input of data for each Paint Product ( <i>including data on each Offering</i> )	5
-----	---	---

- 4) For each Offering of each Paint Product, calculate, store (*in field b-iv*), and display the total profit of the offering. This average must be displayed for exactly 2 decimal places. Profit is calculated by working out the cost (*number of units sold, multiplied by size, multiplied by cost to produce per litre*), subtracting this from the total units sold, multiplied by the selling price per unit.

4.1	Calculation of total profit per Paint Product for each offering	5
4.2	Display of total profit ( <i>must show for 2 decimal places</i> )	3
4.3	Correct	8



- 5) For each Paint Product, calculate, store (*in field d*) the index of the offering that is the least profitable. Use the index to display the size of this offering.

5.1	Calculation of index of the least profitable offering	5
5.2	Display of size of offering	1
5.3	Correct	6

- 6) Define a function called `DivideByTen` which will take in one `Double` value and return one tenth of it.

6.1	Definition of <code>DividebyTen</code> function	2
6.2	Calculation	2

- 7) For each Paint Product, store (*in field e*), and display the Paint Rating. This is calculated by evaluating one tenth of the least profitable offering of the Paint Product (*using the `DividebyTen` function*) as follows (*use a `Select Case`*):

1/10 of least profitable offering profit	Paint Rating
0 to 100	"A"
In between	"B"
1000 and higher	"C"

7.1	Calculate Paint Product Paint Ratings ( <i>using <code>DivideByTen</code> function</i> )	3
7.2	Display of Paint Ratings	1
7.3	Correct	8

- 8) The twins are particularly interested in the success (or lack thereof) of their first product. Determine if there is an increasing trend in number of units sold for the first Offering of the first Paint Product. For example, if Week 1 sold 10 units, Week 2 sold 10 units, and Week 3 sold 20 units, it would not be considered increasing.

8.1	Determine if increasing pattern exists from Week to Week for Product 1, Offering 1	8
8.2	Display	2
8.3	Correct	10

\\(^v^)\*(^v^)/

