



## FACULTY OF SCIENCE

### ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

<b>MODULE</b>	<b>IFM01A1 &amp; IFM1A10</b> INTRODUCTION TO ALGORITHM DEVELOPMENT (VB)
<b>CAMPUS</b>	APK
<b>EXAMINATION</b>	JUNE 2016 (PAPER B)

<b>DATE</b>	2016-06-10	<b>SESSION</b>	08:30 - 11:30
<b>ASSESSORS</b>	MR M CILLIERS DR WS LEUNG		
<b>INTERNAL MODERATOR</b>	MR D COTTERRELL		
<b>DURATION</b>	180 Minutes	<b>MARKS</b>	100

### INSTRUCTIONS

- ✍ This question paper comprises 3 (three) pages (including this cover page)
- ✍ The first 20 (twenty) minutes are reserved for DESIGN
- ✍ Answer ALL questions (a-g)
- ✍ The marksheet on page 3 (three) forms part of the question
- ✍ No further marks will be awarded for Correct Execution from the point your submitted program terminates unexpectedly
  
- ✍ Project naming convention: CXXX\_YYYYYYYYY where X represents your computer number and Y represents your student number (e.g. student 201600001 sitting at computer number 10 will name their project C010\_201600001)
- ✍ 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
  
- ✍ When you are ready to submit: read through page 3 for submission guidelines and 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**



The community of Fun in the Sun have approached you to design and develop a Visual Basic application that will assist the group of volunteer lifeguards with their record keeping. Your solution must reflect good programming practices as taught to you in the course of this module (decent UI design, modularity, commented code, etc.). Where **Double** values must be displayed, use the **Format** command to limit values to 2 decimal places.

- a) The application will need to store the following data about each Area in the form of records:

Field	Example Data
1. Area name	"North Side"
2. The following data for each Lifeguard stationed in the Area:	
i. Years of experience	3
ii. Number of incidences reported each Day	1, 3, ..., 2
iii. Average number of incidences reported	See Question d
3. Index of Lifeguard with highest average number of incidences	See Question e
4. Area rating	See Question f

- b) Users of the application must be able to handle any number of Areas, any number of Lifeguards per Area, and any number of Days per Lifeguard. You may assume that:
- Each Area will have the same number of Lifeguards; and
  - Each Lifeguard will be on duty for the same number of Days.
- c) Ask the user to input the necessary data for each Area (and its Lifeguards) into records.
- d) For each Lifeguard stationed in each Area, calculate, store (in field 2iii), and display the average number of incidences.
- e) For each Area, calculate, store (in field 3), and display the index of the Lifeguard with the highest average number of incidences. This is the Lifeguard with the highest average incidences.
- f) For each Area, calculate, store (in field 4), and display its Area rating. To obtain the Area rating, you will need the Average number of incidences (field 2iii), as well as the Years of experience (field 2i) of the Lifeguard with the highest average number of incidences to calculate the Rating as follows:

When Years of Experience Is...	Area rating is...
Even	Average incidences / 2
Odd	Average incidences / 2.5

Calculating each of the Area ratings should be done with a function called **CalcRate** that accepts a **Double** variable **Ave**, an **Integer** variable **Exp**, and returns the answer in the form of another **Double** variable. In addition, an **If** statement must be used inside **CalcRate**.

- g) The community of Fun in the Sun would like to know if their initiative has helped to create a trend of decreasing incidences. This is determined by evaluating the ratings of the different Areas (field 4). Determine and display the answer in a separate textbox control.

**COMPUTER NUMBER  
CARD**

Student Number:

Surname, Inits /  
Passport / ID:

Submission Checklist (check the blocks that apply):

- I have followed the naming convention as instructed on page 1 ☐
- I have set the **DebugState** of the UJGrid control to **False** ☐
- A zip of the full and final project has been uploaded to Eve ☐
- A backup zip of the full and final project has been:
  - Burnt to CD ☐
  - Saved to USB by my invigilator ☐

or

I further acknowledge that:

- I know that a non-compiling submission is capped at 40%
- I take full responsibility for ensuring that my submission is correct and the version I intend to submit for assessment

Signature:  Invigilator's initials:

Mark Allocation		Total
0.1	Design (Input, Output, Events & Actions, Variables, Record structures, Interface, Algorithms)	5
0.2	Implementation of Interface	2
0.3	Option Statements	1
0.4	Effective use of subroutines	1
0.5	Commenting	1
A.1	Definition of Lifeguard record structure	4
A.2	Definition of Area record structure	5
B.1	Input of number of Areas, Lifeguards, and Days	2
B.2	Set up of dynamic arrays	5
B.3	Set up of UJGrid control, DebugState of UJGrid set to <b>False</b>	2
B.4	Labelling of UJGrid control headings (Areas as Rows, and Lifeguards as Columns)	3
C.1	Input of data for each Area (including data on each Lifeguard)	5
D.1	Calculation of Average Incidences per Lifeguard per Area	5
D.2	Display of Average Incidences per Lifeguard per Area	2
D.3	Correct	7
E.1	Calculation of Index of Highest Average Incidences per Area	6
E.2	Display of Index of Highest Average Incidences per Area	1
E.3	Correct	7
F.1	<b>CalcRate</b> Function (includes <b>If</b> )	4
F.2	Use of <b>CalcRate</b> function in calculation of Area rating	3
F.3	Display of Area rating	1
F.4	Correct	8
G.1	Determination of Decreasing Trend in Area ratings	9
G.2	Display of Decreasing Trend in Area ratings	1
G.3	Correct	10
<b>Total</b>		<b>100</b>