**Pre-Release Material for MYE 2019-20**

A teacher needs a program to record marks for a class of 30 students who have sat three computer

science tests.

Write and test a program for the teacher.

• Your program must include appropriate prompts for the entry of data.

• Error messages and other output need to be set out clearly and understandably.

• All variables, constants and other identifiers must have meaningful names.

You will need to complete these three tasks. Each task must be fully tested.

TASK 1 – Set up arrays

Set-up one dimensional arrays to store:

• Student names

• Student marks for Test 1, Test 2 and Test 3

* Test 1 is out of 20 marks
* Test 2 is out of 25 marks
* Test 3 is out of 35 marks

• Total score for each student

Input and store the names for 30 students. You may assume that the students’ names are unique.

Input and store the students’ marks for Test 1, Test 2 and Test 3. All the marks must be validated on

entry and any invalid marks rejected.

TASK 2 – Calculate

Calculate the total score for each student and store in the array.

Calculate the average total score for the whole class.

Output each student’s name followed by their total score.

Output the average total score for the class.

TASK 3 – Select

Select the student with the highest total score and output their name and total score.

**Solutionof prerelease material WITH input prompts, error traps and error messages**

//Task1

DECLARE StdName[1:30] As STRING

DECLARE Test1[1:30] AS INTEGER

DECLARE Test2[1:30] AS INTEGER

DECLARE Test3[1:30] AS INTEGER

DECLARE Total[1:30] AS INTEGER

DECLARE Sum As INTEGER = 0

FOR Count = 1 TO 30

Output “Please enter the name of student #” , Count

INPUT StdName[Count]

Output “Please enter the marks of test 1 for student #” , Count

INPUT Test1[Count]

WHILE (Test1[Count]<0) or (Test1[Count]>20) DO

OUTPUT “Incorrect value, please enter the value again”

INPUT Test1[Count]

ENDWHILE

Output “Please enter the marks of test 2 for student #” , Count

INPUT Test2[Count]

WHILE (Test2[Count]<0) or (Test2[Count]>25) DO

OUTPUT “Incorrect value, please enter the value again”

INPUT Test2[Count]

ENDWHILE

Output “Please enter the marks of test 3 for student #” , Count

INPUT Test3[Count]

WHILE (Test3[Count]<0) or (Test3[Count]>35) DO

OUTPUT “Incorrect value, please enter the value again”

INPUT Test3[Count]

ENDWHILE

//TASK 2

Total[Count] = Test1[Count]+ Test2[Count]+ Test3[Count]

Sum = Sum + Total[Count]

PRINT StdName[Count], Total[Count]

NEXT Count

ClassAverage = Sum/30

PRINT ClassAverage

//TASK 3

DECLARE HighestScore AS INTEGER = 0

DECLARE BestName AS STRING = “xxxx”

FOR Count = 1 TO 30

IF Total[Count] > HighestScore

THEN

HighestScore = Total[Count]

BestName = StdName[Count]

ENDIF

NEXT Count

PRINT BestName, HighestScore

*Sample Questions*

1 (a) All variables, constants and other identifiers should have meaningful names.

1. Declare the array to store the students’ names.

DECLARE StdName[1:30] As STRING

[1]

 (ii) Declare the arrays to store each student’s marks and total score.

 DECLARE Test1[1:30], Test2[1:30], Test3[1:30], Total[1:30] AS INTEGER

[2]

(b) (i)Show the design of your algorithm to complete Task 1 and Task 2 using pseudocode,

programming statements or a flowchart. Do not include any of the validation checks or

input prompts in your algorithm.

DECLARE Sum As INTEGER = 0

FOR Count = 1 TO 30

INPUT StdName[Count]

INPUT Test1[Count]

INPUT Test2[Count]

INPUT Test3[Count]

Total[Count] = Test1[Count]+ Test2[Count]+ Test3[Count]

Sum = Sum + Total[Count]

PRINT StdName[Count], Total[Count]

NEXT Count

ClassAverage = Sum/30

PRINT ClassAverage [8]

(ii)Comment on the efficiency of your design.

Any relevant comment with regards to efficient code (e.g. single loop) [1]

(c) Show two different sets of student data that you could use to check the validation used in

Task 1. Explain why you chose each data set.

Set 1: 20, 25, 35

Reason: valid data to check that data on the upper bound of each range check is

accepted

Set 2: 21, 26, 36

Reason: invalid data to check that data above the upper bound of each range check is

Rejected

 [2]

(d) (i) Explain how you select the student with the highest score (Task 3). You may include

pseudocode or programming statements to help illustrate your explanation.

Description (max 3)

– set variable called HighestScore to zero and variable called BestName to dummy

value

– loop 30 times to check each student’s total score in turn

– check student’s score against HighestScore

– if student’s score > HighestScore then

– … replace value in HighestScore by student’s score and store student’s name in

BestName

– output BestName and HighestScore outside the loop

Sample algorithm (max 3):

DECLARE HighestScore AS INTEGER = 0

DECLARE BestName AS STRING = “xxxx”

FOR Count = 1 TO 30

IF Total[Count] > HighestScore

THEN

HighestScore = Total[Count]

BestName = StdName[Count]

ENDIF

NEXT Count

PRINT BestName, HighestScore

If algorithm or program code only, then maximum 3 marks [5]

(ii) How does your program work when there is more than one student having the highest

score? Explain using your method given in part (d)(i).

comment on which student(s)’ name will be output

e.g. The first student with the highest score will be output [1]