GCE AS/A LEVEL

2500U20-1

COMPUTER SCIENCE – AS unit 2
Practical Programming to Solve Problems

FRIDAY, 24 MAY 2019 – MORNING

2 hours

INSTRUCTIONS TO CANDIDATES

Answer **ALL** questions.
The question paper is divided into two main sections. Below are the recommended timings for this assessment.

Section A
You are advised to spend no more than **60 minutes** on this section.
You will be required to analyse and deconstruct the scenario to consider its component parts in terms that can be addressed through automated computation.

Section B
You are advised to spend no more than **60 minutes** on this section.
You will be required to develop programs that solve computing problems referring to the initial scenario.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question.

The total number of marks available is 60.

Assessment will take into account the quality of written communication used in your answers.

You will need a computer with an installed functional copy of the Integrated Development Environment (IDE) appropriate to your chosen programming language and word processing software.

A calculator is allowed in this examination.

Remember to save your work regularly.

Scenario

KimsKarate

KimsKarate is a karate training club based in Croesyceiliog. It offers training sessions for different age and ability groups.

KimsKarate employs several instructors and currently records its data on paper.
KimsKarate wants a computerised system, which will allow the club to store details of its instructors, the groups they take and the pupils enrolled in each group.

Currently a single instructor can be responsible for many groups, and a group has many pupils. Pupils can only be enrolled in one group at a time and there is always a single instructor responsible for that group.

The main requirements of the new computer system for *KimsKarate* are:

• 	The ability to store details on instructors, groups and pupils.
• 	The ability to calculate pupils' grades based on test results.
• 	The ability to accept electronic payment for training sessions in the future.
• 	The ability to carry out simple calculations, such as to calculate the total price of several session fees.

KimsKarate intends to use the following tables to store data:

Instructor Table	**Group Table**	**Pupil Table**

Section A: Analysis and Design (36 marks)

You have been asked to analyse the scenario as a preliminary step towards creating a prototype computer system for *KimsKarate*.

Answer all questions.

Present your answers as a single word-processed document named *Section A Analysis and Design*.

1. Create an entity relationship diagram for this scenario. [6]

2. Draw **two** data structure tables that will allow *KimsKarate* to store instructor details and pupil details. [10]

3. Select and fully justify your proposed method of solution for the three main requirements listed in the scenario. [6]
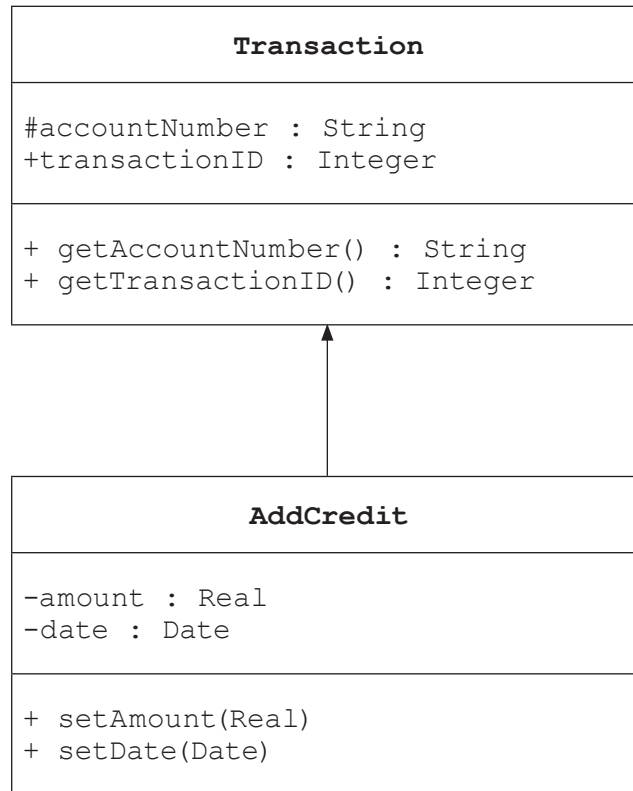
4. Below is a flowchart which helps instructors determine if a pupil has gained enough marks in 3 tests to attain a pass.
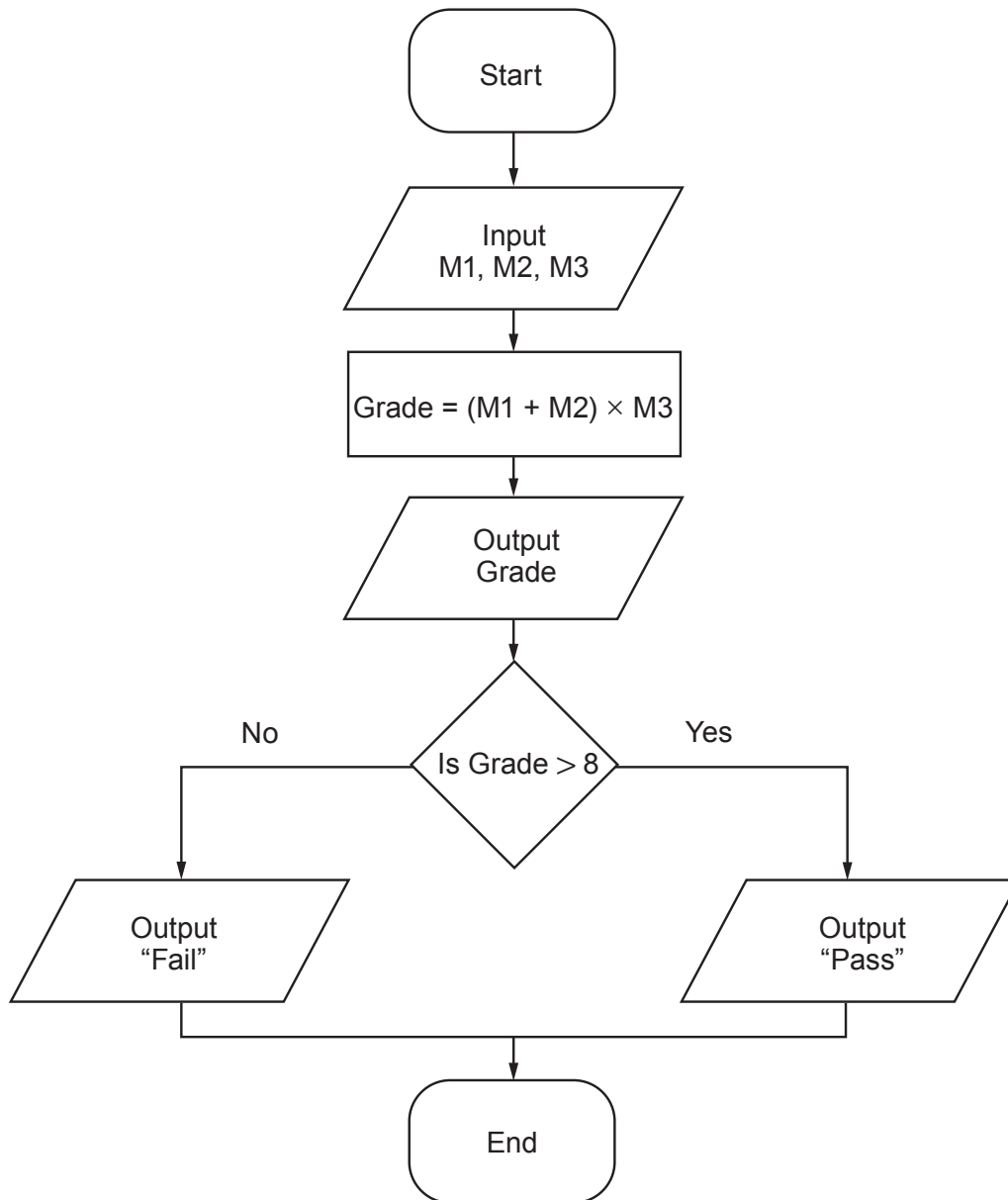
Copy and complete the table below to show the outputs for the flowchart with the inputs given.

[8]

M1	M2	M3	Outputs
1	2	3	
3	2	1	
2	1	3	
3	1	2	

5. *KimsKarate* has previously commissioned a systems analyst to prepare a class diagram to allow it to implement a payment processing system. The class diagram below represents the relationships between some of the classes that they could use.

```
┌─────────────────────────────────────────┐
│              Transaction                 │
├─────────────────────────────────────────┤
│ #accountNumber : String                  │
│ +transactionID : Integer                 │
├─────────────────────────────────────────┤
│ + getAccountNumber() : String            │
│ + getTransactionID() : Integer           │
└─────────────────────────────────────────┘
                     ▲
                     │
┌─────────────────────────────────────────┐
│               AddCredit                  │
├─────────────────────────────────────────┤
│ -amount : Real                           │
│ -date : Date                             │
├─────────────────────────────────────────┤
│ + setAmount(Real)                        │
│ + setDate(Date)                          │
└─────────────────────────────────────────┘
```

(a) State the name of an identifier for a subclass. [1]

(b) State the name of an identifier for a superclass. [1]

(c) An object called `transact1` is instantiated from the class `AddCredit`.

   (i) Applying the concept of inheritance, list **all** the methods that would be contained within the object `transact1`. [2]

   (ii) Applying the concept of inheritance, list **all** the attributes that would be accessible within the object `transact1`. [2]

Section B: Develop programs (24 marks)

Select the programming language of your choice from section BI, BII or BIII and answer both questions within your chosen section.

BI Visual Basic

KimsKarate **wants a prototype computer system to be developed using Visual Basic.**

1. **Open the file *Pupils***
    *   Read through the code and familiarise yourself with its contents
    *   The file contains code which can save pupil details and return the number of pupils on file that match the search criteria entered e.g. First name.

    Using the internal facility of your chosen language, **annotate** the code provided to clearly explain the design of this program to another software developer.

    Save the changes made to the file *Pupils*.   [8]

2. The staff of *KimsKarate* frequently need to perform quick, simple calculations.
    Create a new form that will allow *KimsKarate* to:

    *   Input integer numbers 0 to 9
    *   Carry out addition (+) on two numbers
    *   Carry out subtraction (-) on two numbers
    *   Carry out multiplication (x) on two numbers
    *   Display the result of the operation on the screen
    *   Store the result in a text file called lastResult.txt
    *   Recall the last value from the text file called lastResult.txt
    *   Have an option to clear the screen of previous results

    Save your work as *KimsCalc*   [16]

BII Java

KimsKarate **wants a prototype computer system to be developed using Java.**

1. **Open the file** *Pupils*
   - Read through the code and familiarise yourself with its contents
   - The file contains code which can save pupil details and return the number of pupils on file that match the search criteria entered e.g. First name.

   Using the internal facility of your chosen language, **annotate** the code provided to clearly explain the design of this program to another software developer.

   Save the changes made to the file *Pupils*. [8]

2. The staff of *KimsKarate* frequently need to perform quick, simple calculations.
   Create a new form that will allow *KimsKarate* to:

   - Input integer numbers 0 to 9
   - Carry out addition (+) on two numbers
   - Carry out subtraction (-) on two numbers
   - Carry out multiplication (x) on two numbers
   - Display the result of the operation on the screen
   - Store the result in a text file called lastResult.txt
   - Recall the last value from the text file called lastResult.txt
   - Have an option to clear the screen of previous results

   Save your work as *KimsCalc* [16]

BIII Python

KimsKarate **wants a prototype computer system to be developed using Python.**

1. **Open the file *Pupils***
   - Read through the code and familiarise yourself with its contents
   - The file contains code which can save pupil details and return the number of pupils on file that match the search criteria entered e.g. First name.

   Using the internal facility of your chosen language, **annotate** the code provided to clearly explain the design of this program to another software developer.

   Save the changes made to the file *Pupils*.   [8]

2. The staff of *KimsKarate* frequently need to perform quick, simple calculations.
   Create a new form that will allow *KimsKarate* to:

   - Input integer numbers 0 to 9
   - Carry out addition (+) on two numbers
   - Carry out subtraction (-) on two numbers
   - Carry out multiplication (x) on two numbers
   - Display the result of the operation on the screen
   - Store the result in a text file called lastResult.txt
   - Recall the last value from the text file called lastResult.txt
   - Have an option to clear the screen of previous results

   Save your work as *KimsCalc*   [16]

END OF PAPER