General Certificate of Education
Advanced Subsidiary Examination
June 2011

Computing COMP1/PM

Unit 1 Problem Solving, Programming, Data Representation and Practical Exercise

Preliminary Material

To be given to candidates on or after Tuesday 1 March 2011, subject to the instructions given in the *Teachers' Notes* (COMP1/TN).

Information

- This Preliminary Material comprises
  - Instructions to Candidates
  - a Data File.

- A Skeleton Program is provided separately by your teacher and must be read in conjunction with this Preliminary Material.
- Candidates are advised to familiarise themselves with the Preliminary Material and Skeleton Program before the examination.
- This Preliminary Material will be made available to you again in the examination. You will also be given access to the Skeleton Program electronically at the start of the examination. You must **not** take any copy of the Preliminary Material, Skeleton Program or any other material into the examination room.

Instructions for Candidates

The question paper is divided into four sections and a recommendation is given to candidates as to how long to spend on each section. Below are the recommended timings for the 2011 examination.

Section A
You are advised to spend no more than **30 minutes** on this section.
Questions will examine the specification content **not** specific to the **Preliminary Material**.

Section B
You are advised to spend no more than **20 minutes** on this section.
You will be asked to create a new program **not** related to the **Preliminary Material** or **Skeleton Program**.

Section C
You are advised to spend no more than **20 minutes** on this section.
Questions will refer to the **Preliminary Material** and the **Skeleton Program**, but will not require programming.

Section D
You are advised to spend no more than **50 minutes** on this section.
Questions will use the the **Skeleton Program** and the **Preliminary Material** and may require the **HiScores.txt Data File**.

Electronic Answer Document

Answers to questions for all four sections must be entered into the word processed document made available to you at the start of the examination and referred to in the question paper rubrics as the **Electronic Answer Document**.

Preparation for the Examination

You should ensure that you are familiar with this **Preliminary Material** and the **Skeleton Program**.

For the Skeleton Program for your programming language, you should be familiar with:

- the built-in functions available for manipulating string data and converting strings to other data types
- file handling commands for CSV (Comma Separated Variable) files
- declaring and using arrays.

Dice Cricket Game

The **Skeleton Program** is a program for the two-player game of Dice Cricket. Dice Cricket is a simple game based on the sport of cricket.

When playing Dice Cricket, players use two special dice called the Bowl Die[1] and the Appeal Die. The Bowl Die is a 6-sided die where, instead of the numbers 1 to 6, the sides have "0", "1", "2", "4", "6", and "OUT" written on them. The player whose turn it is rolls the Bowl Die. If the result is one of the numeric values then this is added to their score (the number of *runs* they have got) and they continue to roll the Bowl Die until "OUT" is rolled. If the result is "OUT" then they roll the Appeal Die to see what happens next.

The Appeal Die is a 4-sided die where the numbers 1 to 4 have been replaced with different values. The values on the Appeal Die are "NOT OUT", "CAUGHT", "LBW" and "BOWLED". If when a player rolls the Appeal Die they get a result of "NOT OUT" then their turn continues and they can roll the Bowl Die again. Any other result on the Appeal Die means that they are out and their turn finishes. When player one's turn is over player two has their turn. When player two is out the two players' scores are compared and the winner is the one with the highest score.

In the **Skeleton Program** players can choose to play with real dice or virtual dice. If real dice are used then the two players have actually got a Bowl Die and an Appeal Die and enter the values they roll into the program. If virtual dice are used then the program simulates the rolling of the Bowl Die and Appeal Die by generating random numbers. There are 6 different values on the Bowl Die and 4 different values on the Appeal Die.

The **Skeleton Program** also stores the names and scores of the four highest results obtained by players playing Dice Cricket. After each game the scores of the two players are compared with the previous top scores. The winner's details, if their score is higher, will replace those of the player with the lowest top score. The loser's details, if their score is high enough, could also replace one of the previous player's details. If the winning player's score is the same as the lowest top score their details are not stored. If a game is drawn with a score lower than three of the top four scores then only player one's details will be stored.

In the **Skeleton Program** there is a menu containing five options:

- Play game version with virtual dice
- Play game version with real dice
- Load top scores
- Display top scores
- Quit

If the user chooses to load top scores from a file then the file **HiScores.txt** is opened and the contents placed in the array `TopScores`.

The Data File

```
Ricky,12
Sachin,45
Brian,2
Monty,1
```

The data file **HiScores.txt** will be available to you at the start of the examination.

[1] *Dice* is the plural of the word *die*.

Variables

Some of the variables used are:

Identifier	Data Type	Purpose
`TopScores`	`Array[1..4] of Record`	This is an array of records. Each record stores the details about one of the top four scores. A record consists of a `name` (`string` data type) and a `score` (`integer` data type)
`PlayerOut`	`Boolean`	Used to indicate if a player is out or not
`CurrentPlayerScore`	`Integer`	Used to store the number of runs that the current player has accumulated
`PlayerNo`	`Integer`	Used to indicate if it is player one's or player two's turn

Notes

The programming language used to code the game will determine the letter case for each identifier and so may not match exactly the identifiers shown in the table above.

Your chosen programming language may use arrays with a lower bound value of 0. If so, position 0 will not have been used.

END OF PRELIMINARY MATERIAL