OCR
RECOGNISING ACHIEVEMENT

Released June 2012 For Assessment Submission January 2013 to June 2015

GCSE COMPUTING

A453/01 Programming Project

CONTROLLED ASSESSMENT MATERIAL 3

This assessment may be periodically reviewed. Please check on OCR Interchange that you have the Controlled Assessment material valid for the appropriate assessment session.

A45301

A433330612

INSTRUCTIONS TO TEACHERS

- Please refer to Section 4 of the GCSE Computing specification for instructions on completing this controlled assessment task.
- Each task can be contextualised appropriately to suit facilities available in your centre.
- The marking criteria should be available to candidates whilst completing the tasks.
- The quality of written communication will be assessed in the testing section.
- The total number of marks for this unit is **45**.

INFORMATION FOR CANDIDATES

- This document consists of **4** pages. Any blank pages are indicated.

Teachers are responsible for ensuring that assessment is carried out against the Controlled Assessment set for the relevant examination series (detailed above).

Assessment evidence produced that does not reflect the relevant examination series will not be accepted.

Turn over

This assessment consists of three tasks.

Candidates should complete all tasks.

The tasks are set so as to enable all the techniques identified in the specification to be demonstrated in their solution. The tasks provide opportunities to demonstrate a range of skills and all three tasks contribute to the overall mark awarded for this assessment. Marks are awarded for using the appropriate skills and techniques effectively and efficiently to produce a solution to these three tasks. Not all techniques will be required for each of the subtasks.

Task 1 Simulating a dice

A game uses dice with 4, 6 and 12 sides to determine various outcomes.

Design, code and test a program that will simulate throwing dice with these numbers of sides.

The user should be able to input which dice is being thrown, eg 4, 6 or 12.

The program should output the dice chosen and the score,
for example '6 sided dice thrown, score 4'

The user should be able to repeat this process as many times as required.

Task 2 Determining a character's attributes

When determining certain characteristics of a game character the numbers on a combination of dice are used to calculate certain attributes.

Two of these attributes are strength and skill.

At the start of the game, when the characters are created, a 4 sided dice and a 12 sided dice are thrown to determine values for each of these attributes using the following method for each character:

- Each attribute is initially set to 10.
- The score on the 12 sided dice is divided by the score on the 4 sided dice and rounded down.
- This value is added to the initial value.

This process is repeated for each attribute for each character.

Describe this process using a suitable algorithm.

Write and test the code to determine these two attributes for a character and store the sample data for two characters, including suitable names, in a file.

Task 3 Determining the outcome of an encounter

When there is an encounter between two characters the outcome is determined by the following process:

- The differences between the strength attributes for the two characters is calculated
- This difference is divided by 5 and then rounded down to create a 'strength modifier'
- The process is repeated for the skill attribute to create a 'skill modifier'
- Each player throws a 6 sided dice.
    - If the scores on both dice are the same, no changes are made
    - If the scores are not the same, the player with the highest score adds the 'strength modifier' to the strength value and the 'skill modifier' to the skill value for their character
    - The player with the lower score on the dice subtracts these modifiers from the strength and skill values for their character
- If a skill value becomes negative, then it is stored as zero
- If a strength value becomes zero or negative, then the character dies.

The program should:

- Allow the user to input the strength and skill for two characters.
- Display the outcome of the encounter using the process above.

Design an algorithm to describe this process. Write, test and evaluate the code.