University of Westminster, Department of Computer Science

|  |  |
| --- | --- |
| **ECSC410 Software Dev Principles 1: Project 1 - Guess the Word Game** | |
| Module leaders | Guhanathan Poravi, Aloka Fernando, Pumudu Fernando, Achala Aponso |
| Unit | Project 1 |
| Weighting: | 35% of the module |
| Qualifying mark | You must get 30 marks or above for this final piece of work |
| Description | Guess the Word Game |
| Learning Outcomes: | LO3 LO4 LO5. |
| Handed Out: | 07th of Nov 2015 |
| Due Date | * Code due on Blackboard coursework upload by Monday 14th of Dec 2015 10am. |
| Expected deliverables | Java program code upload, plus documentation |
| Method of Submission: | Blackboard + In-class test |
| Type of Feedback and Due Date: | Your in-class test mark (worth 70%) and java code mark (worth 30%) should appear on Blackboard Gradebook within 3 weeks of the test. Individual written feedback will be available via the Blackboard ‘Assessment’ link once your marks are shown in gradebook. If you would like extra feedback about your grade, or your mark does not show, then please contact your project tutorial tutor.  **All marks will remain provisional until formally agreed by an Assessment Board.** |

**Assessment regulations**Refer to section 4 of the “How you Study - A guide for Undergraduate Students” for a clarification of how you are assessed, penalties and late submissions, and what constitutes plagiarism, etc.

**Penalty for Late Submission**If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website: [**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**](http://www.westminster.ac.uk/study/current-students/resources/academic-regulations)

All coursework code on this module is submitted via Blackboard. It may be automatically scanned through a text matching system (designed to check for possible plagiarism).

• You DO NOT need to attach a copy of the CA1 form;

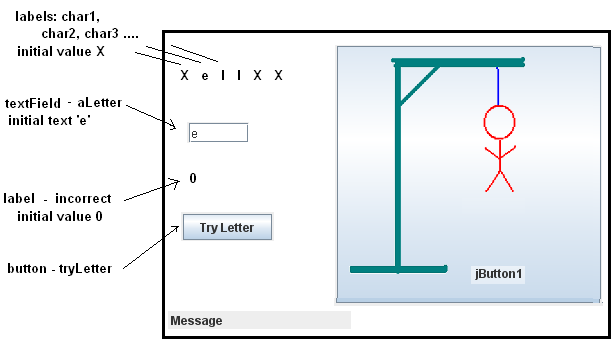
To submit your assignment:

• Log on to Blackboard at http://learning.westminster.ac.uk; and follow the instructions below.

If you are unable to submit your work on Blackboard due to a finance hold you must email your work to FSTRegistry@westminster.ac.uk by the same deadline, putting on the subject line the module code, assessment number, and your name. This shows that you have completed your work by the deadline. After the finance hold is lifted you must then submit the same work as normal on Blackboard, otherwise it will not be marked and you will get a fail for that part of the assessment.

**Coursework Description:**  **Project 1 - Guess the Word Game**

This **exercise** is to create a hang man type program. If you do not like the idea of a hang man image then use a different sequence of images (such as pushing someone out of the door bit by bit) as an alternative.



Your program should have a hidden word stored internally. (Initially use the word ‘hello’)

The ‘X’s in the labels char1, char2 etc., will represent the characters of the hidden word. (The first X represents ‘h’, the second X represents ‘e’, so all the characters of ‘hello’ are shown as Xs)

The program user will guess one letter at a time (and type it into the textField and click on the button), and if they guess a letter which is in the hidden word then the appropriate ‘X’ is replaced by the correctly guessed letter. The diagram shows that ‘e’ and ‘l’ have been guessed correctly.

Every time an incorrect letter is guessed then another part of the sequence of the hang man (or alternative) images is shown.

Follow the ‘Guess the Word’ start guide on Blackboard to help you create the application. It is important that your program will work with any 5 or 6 letter word written once at the start of your program code (and not just the word ‘hello’)

Initially, create the full final picture image, and then rub out a piece and save it as a different filename, then rub out another part and save it as a different name, etc. Use the images in your program in the reverse order to how you created the images, so the sequence will gradually build up to the full image being shown.

**Code Extras.**

1. Work out a way to make the program display an appropriate ‘Game Over’ message. For example, the message could display ‘Congratulations you have won. Game over. ’ when the hidden word has been guessed before the final complete hang man (or alternative) image is displayed, or ‘You did not win. Game over.’ when the word has not been guessed before the final image is displayed.

2. Make the program work for any length hidden word from 3 to 10 characters, and only display sufficient ‘X’s to match the length of the hidden word.

3. Write 6 different hidden words into your program and make it chose a random word each time.

4. Read in a hidden word from a text file of words. The program should choose a random word from the file.

**Submission Instructions:**

Java Program code to be submitted as a single text file by the date shown above on Blackboard ECSC410 ‘Submit Coursework’ link. For this submission you should copy and paste all your code into ONE long text file, type the filename at the top of each piece of pasted code and write whether it is part 1 or 2 or 3 or 4 , and at the very top of the text file type your name and Unix account number. Call this large saved file by your ID number but start with GW (e.g. . GWw123456.txt ) and ensure it is saved as plain text (not .doc or .zip). Do not change your code after it has been submitted.

If Blackboard doesn’t work at the last minute then upload the file within 24hrs of Blackboard working again. You do not need to email us as we will know Blackboard is not working. Be careful and double check your file as you can only upload once.

We will give feedback via Blackboard (see the assignment spec shown under the 'Java Project Materials' link).

**Coursework Marking Scheme**

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Weighting |  | Comments |
| Code solution | 30% | 21 marks will be given for the code solution if you have submitted a reasonable solution without the extras mentioned above. You can achieve a maximum of 30 marks depending on how many of the code extras you have implemented. | |
| In-class test | 70% | 20 MCQs based on Lecture 01 to 10 except file handling | |