Assignment 2 Mark Scheme

* Clearly documented program code – 15 marks
  + Start with full marks
  + Lose a mark for each variable not accompanied by a comment explaining its purpose
  + Lose a mark for each loop, procedure or function without an explanation of what it does
* Excellent screen layout – 15 marks
  + Meaningful input message – 1
  + Lives left – 2
  + Incorrect guesses – 2
  + Word so far – 4
  + Interface is presentable and easily interpreted - 6
* Effective code layout – 15 marks
  + Start with full marks
  + Lose ½ mark for each uninformative or misleading variable and function name up to 5 marks
  + Lose ¼ for each line of code without proper indentation up to 2.5 marks
  + Lose ¼ mark for each other break with the Google C++ style guide up to 7.5 marks
* Accurate programming – 30 marks
  + Program requirements - 26
    - Accepts user input - 1
    - Correctly tracks correct guesses - 6
    - Correctly tracks incorrect guesses - 6
    - Correctly tracks number of lives - 3
    - Loads word from file – 4
    - Select word from file randomly – 4
    - Informs user of final game outcome - 2
  + Duplicated code effort – starts with 2
    - Lose ½ mark for each instance of needless duplications
  + Minimized processing and memory consumption – starts with 2
    - Lose ½ mark for each instance of using more memory than necessary when it doesn’t increase readability up to 5 marks
    - Lose ½ mark for each instance of using more processing than necessary when it doesn’t increase readability up to 5 marks
* Appropriate testing – 15 marks
  + Handle situation where load dictionary doesn’t exist – 2 marks
  + Make sure a valid word is always selected from the dictionary – 2 marks
  + Character entered is a letter – 3 marks
  + Character has not been entered already – 3 marks
  + Input is not case sensitive e.g. entering lower case ‘a’ and then upper case ‘A’ should count as the same input – 3 marks
  + Meaningful error messages – 2 marks