

# WORKSHEET A: INTRODUCTION TO C++

Version 1.0  
BSc Computing for Games  
COMP130

Dr Ed Powley

## Introduction

The open-world RPG **Fallout 4** contains a terminal hacking minigame, in which the player must guess a secret  $n$ -letter word. In COMP110 Worksheet B, you modelled this minigame using a flowchart and a piece of pseudocode. In this worksheet, you will implement the minigame as a C++ console application.

In the minigame, each guess receives a **likeness** score, defined as the number of letters which match the secret word (i.e. the same letter in the same position). For example if the secret word is HOUSE and the guess is MOUSE, the likeness is 4 out of 5. If the guess is HOPES, the likeness is 2 out of 5 (the letters S and E do not count as they are in the wrong positions). The minigame ends when the player guesses correctly, or after four incorrect guesses. (In Fallout 4 there is a way to replenish the number of available guesses, but this is disregarded in this worksheet for simplicity.)

---

*"War. War never changes."*  
— The Narrator, *Fallout*

---

Algorithm 1 gives pseudocode for calculating the likeness between a guessed word and the secret word. Figure 1 shows a flowchart for the overall game.

To complete this worksheet:

- (a) **Implement** Algorithm 1 as a C++ function. The function should take two `std::strings` as arguments, and return an `int`.
- (b) **Complete** the provided template program so that it implements the logic shown in Figure 1.

---

**Algorithm 1** An algorithm for calculating the likeness score for the terminal hacking minigame.

---

```
procedure GETLIKENESS(guessedWord, secretWord)
    result ← 0
    for  $i = 0, 1, \dots, \text{secretWord.length} - 1$  do
        if  $\text{secretWord}[i] = \text{guessedWord}[i]$  then
            increment result
        end if
    end for
    return result
end procedure
```

---

## Submission instructions

Begin by **forking** the GitHub repository at the following URL:

<https://github.com/Falmouth-Games-Academy/comp130-worksheets>

The `worksheet_A` directory contains a Visual C++ project to build upon. **Read** and **understand** the provided code. **Add** code to satisfy the requirements listed above. You may make any changes to the provided code that you see fit, and may add new files to the project, however please do **not** move or



The terminal hacking minigame in *Fallout 4*.

rename the project or the existing files.

Attend the timetabled COMP130 session in **week commencing 23rd January 2017**, ensuring that you have uploaded all material to GitHub before this time.

## Marking criteria

Remember that **it is better to submit incomplete work than to submit nothing at all**. Any attempt, even unfinished, will receive a passing grade.

Your work will be marked according to the following criteria:

- **Functional coherence.** Is your implementation correct and free of obvious bugs? Does it adhere to the specification outlined in this worksheet?
- **Sophistication.** Have you made use of appropriate code structures and data structures? Note the emphasis is on **appropriate**; extra credit will **not** be given for unnecessarily complex solutions.
- **Maintainability.** Is your code well commented? Are your identifier names appropriate and descriptive? Have you adhered to appropriate coding standards?

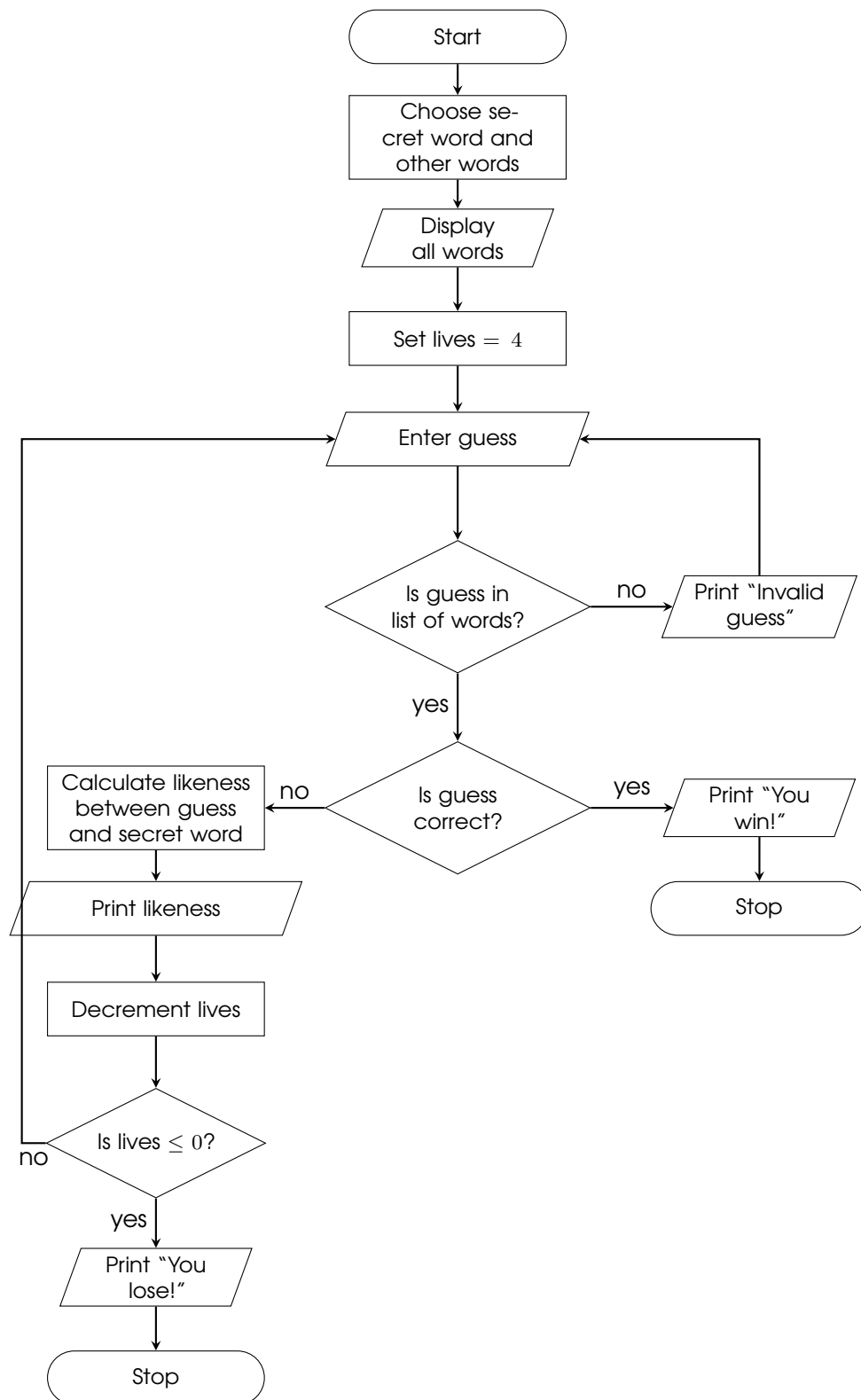


Figure 1: Flowchart for the Terminal Hacking game