University of Glasgow | School of Computing Science

# Assessed Coursework

| | | | | |
|---|---|---|---|---|
| **Course Name** | **Advanced Systems Programming (H/M)** | | | |
| **Coursework Number** | **Exercise 1** | | | |
| **Deadline** | **Time:** | **4:30pm** | **Date:** | **25 February 2026** |
| **% Contribution to final course mark** | **10%** | | | |
| **Solo or Group** ✓ | **Solo** | ✓ | **Group** | |
| **Anticipated Hours** | **10** | | | |
| **Submission Instructions** | **Submit via Moodle** | | | |
| **Please Note: This Coursework cannot be Re-Assessed** | | | | |

## Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below.

The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline

a. the work will be assessed in the usual way;

b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

**Penalty for non-adherence to Submission Instructions is 2 bands**

# Advanced Systems Programming (H/M) 2025-2026 – Exercise 1

School of Computing Science, University of Glasgow

11 February 2026

## Introduction

Programming languages and systems have traditionally used one of three approaches to managing memory: the system either provides a garbage collector, or it supports reference counting with automatic memory reclamation when the reference count reaches zero, or it relies on the programmer to manually allocate and free memory. There are advantages and disadvantages to each, but the fact that all three are widely used suggests that none of them are suitable for all problem domains.

The Rust programming language uses an alternative approach, known as *region-based memory management*. This aims to provide effective automatic memory management, without the indeterminism and overheads of a garbage collector, by tracking ownership of data and using this understanding of ownership to automatically deallocate objects when they go out of scope.

## Assessed Exercise 1

**In this assessed exercise**, you will study region-based memory management in Rust and prepare a short written paper describing how it works, how it compares to manual memory management in other languages, and how it affects what programs can easily be written in Rust. There are four parts to this exercise:

1. Research and prepare a description of how memory management works in the Rust programming language, and how the Rust language manages ownership of data, including different pointer types and borrowed references to data. Describe when memory is allocated and deallocated in Rust programs, how Rust tracks ownership and borrowing of data and how this is controlled via restrictions on references, and what safety guarantees result from this control. **[10 marks]**

2. Compare memory management in Rust with memory management in the C programming language. Discuss how memory management in these two languages compares in terms of 1) programmer effort to manage memory; 2) runtime efficiency of memory management; and 3) safety and flexibility of memory management. **[10 marks]**

3. Compare Rust's memory management to garbage collection, highlighting the relative advantages and disadvantages of the two approaches. **[4 marks]**

4. With the aid of sample code fragments, discuss both the types of program that Rust's approach to memory management makes easy to write, and the types of program that are difficult, or impossible, to write in Rust. Explain what language design decisions make such programs easy or difficult to write. **[6 marks]**

## Submission

You should submit a single written paper, in PDF format, that contains the answers to the four parts of the assessed exercise. A mark out of 30 will be assigned to your submission, weighted as noted earlier. This mark will be converted to a percentage, then used to assign a band on the University's 22 point scale.

Prepare your PDF file formatted for A4 paper, in two columns, using the Times Roman font in 10pt, with 1.5cm margins. Ensure the exercise title, your GUID, and the date are included at the top of the first page (i.e., use a format similar to this page of the handout). If you use LaTeX to prepare your report, your `.tex` file should start like the following (you are required to use LaTeX):

```
\documentclass[10pt,a4paper,twocolumn]{article}
\usepackage[cm]{fullpage}
\usepackage{newtxtext}
\usepackage{newtxmath}
\begin{document}
\title{Advanced Systems Programming (H/M) Exercise 1}
\author{1234567a}  % Replace with your GUID
\date{\today}
\maketitle
\section*{Part 1}
```

Your report must not exceed four pages in length, including all figures, tables, code samples, and references. Length is not an indication of merit, and if you can complete the assignment in less than four pages, then you are encouraged do so.

You must submit your report before 4:30pm on 25 February 2026. Following the code of assessment, late submissions will be accepted for up to 5 working days beyond this due date. Late submissions will receive a two band penalty for each working day, or part thereof, the submission is late. Submissions that are received more than five working days after the due date will be awarded a band of H. If you are ill, or have other circumstances that may affect your submission, then you may contact the student support office *before* the deadline to request an extension, following the usual procedure.

Submissions must be made via Moodle. Submit a single PDF file entitled `asp-ex1-`*`GUID`*`.pdf`, replacing *`GUID`* with your GUID (your student number followed by the first letter of your surname). Submissions that do not follow submission instructions will be given a two band penalty. Penalties will be strictly enforced.

**This assessed exercise is worth 10% of the mark for this course**. It is not expected to take more than ten hours to complete, including both the time spent on background reading and research and the time spent writing up your answers in the form of a paper.

Feedback will comprise a summary of factors considered when marking, a breakdown of the marks awarded, and brief written comments.