

CS3004 Network Computing

Assessment/Coursework for 2023/24

Provisional

TABLE OF CONTENTS

Main Objective of the Assessment.....	1
Description of the Assessment.....	1
Learning Outcomes and Marking Criteria.....	1
Format of the Assessment.....	1
Submission Instructions.....	1
Avoiding Academic Misconduct.....	2
Expectations of Artificial intelligence Use.....	2
Late Coursework.....	2

Assessment Title	CS3004 Network Computing
Module Leader	Simon Taylor
Distribution Date	10-10-23
Submission Deadline	01-12-23
Feedback by	15-12-23
Contribution to overall module assessment	Threshold-based assessment (see below)
Indicative student time working on assessment	35 Hours
Word or Page Limit (if applicable)	3000 Words/Pages (not including references) suggested
Assessment Type (individual or group)	Individual

MAIN OBJECTIVE OF THE ASSESSMENT

The main objectives of this assessment are to create and report on a network computing application as described below. These objectives satisfy all the learning outcomes of the module as re-stated below:

- Demonstrate a thorough understanding of the main issues related to network computing
- Critically evaluate requirements and problems that arise when advanced network computing applications are designed and implemented
- Demonstrate practical ability in implementing an advanced network computing application

DESCRIPTION OF THE ASSESSMENT

Read carefully the following scenario then follow the instructions given to complete the task.



The WLFB Car Park Application

Scenario and Instructions for Completing the Task

The primary goal of this assignment is to develop a demonstrator of a multi-threading client-server system that shows how a car park management system can keep track of the number of available spaces as cars leave and enter a car park.

The WLFB Car Park

The WLFB Car Park (WCP) has two entrances (EntA and EntB) and two exits (ExA and ExB). A car will attempt to enter the car park and will be allowed to enter if there is space. If not it will queue at the entrance until there is space – the entrances are very dumb and will only check this when another car arrives (this will make the coding a lot easier!) A car leaving the car park will reduce the number of cars in the car park by one. For purposes of this assignment, you may assume that the car park has five spaces (to make testing easier!) The car park entrances and exits are as follows:

Entrances A and B (EntA and EntB)

Each is implemented as a SEPARATE CLIENT. Each has its own queue. Note that this just needs to be an integer (don't use an actual queue!)

Check_space – this returns the current amount of space in the car park.

Add_car – this removes one car from the entrance queue and instructs the server to add one car to the car park.

Exits A and B (ExA and ExB)

Each is implemented as a SEPARATE CLIENT.

Remove_car – this instructs the server to remove one car from car park.

Server

This must be able to represent the number of cars in the car park (as an integer) and be able to respond to the clients appropriately. For the entrances it must be able to inform them if there is space or not. For all clients it must be able to change the number of cars in the car park appropriately and confirm this.

You are required to create a multi-threaded client-server system that uses locking and has:

- Four clients representing the two entrances and two exits implementing the operations as above;
- One server that holds the car park variables and executes the operations as instructed by the clients

You are not required to implement any kind of user interface (so don't!) and you will not get any credit for it (you have other assignments to work on).

NOTE THAT DUE TO PROBLEMS IN PREVIOUS YEARS YOU MUST DEVELOP THIS IN JAVA USING THE SOCKET AND THREAD APIs AS PRESENTED IN TUTORIALS (i.e. ACTIONSERVER) YOU MAY REUSE THE CODE IN THE TUTORIALS AS A BASIS FOR YOUR WORK.



FORMAT OF THE ASSESSMENT

Students are expected to submit one report individually. The format is as follows:

Title page:

Be sure to include your student ID number, the title of the assignment, the name, module code and year of the module and the submission date.

Body:

You must use the following format:

1. Introduction.
 2. Requirements
 3. Design (Architecture and Protocol Sequence)
 4. Implementation
 5. Testing
 6. Conclusions
- Appendix A: Code

The sections are as follows:

Introduction

This section gives a short overview of your assignment.

Requirements

Briefly specify the requirements of your program with a short justification of how these relate to the assignment.

Design

This should be followed by the design discussion. This should capture the overall architecture of your system (the processes and their communication links) and the design of your program that clearly shows relevant details of the **protocol** (the sequence of messages between the clients and the server and message format). You can use one of the techniques in the design and implementation lectures to document this (e.g. algorithm, protocol table, etc.) It is critical that this section discusses how you created the design of this network computer application so that the marker can understand the approach that you took.

You will need to show:

- The protocol between one entrance client and the server.
- The protocol between one exit client and the server.

Implementation

Using code snippets discuss how you implemented the design. Rather than being a description of the code, you should concentrate on how key elements of the design were implemented. The purpose of the section is to demonstrate that you understand how a network computing application can be implemented.



Testing

Show the following in your testing section (supported by screen shots THAT CAN BE EASILY READ):

- EntA, EntB and Server:
 - Successful execution of Check_space at both clients and server (that shows the behaviour when the car park is empty and full.
 - Successful execution of Add_car at both clients and server
- ExA, ExB and Server:
 - Successful execution of Remove_car at both clients and server

Remember that these should show the connection, the protocol, correct locking and updates of relevant values.

Conclusions

A short summary to close your report.

Appendix A

A formatted listing of all your code with a short commentary describing each class.

Word Limit:

The suggested word limit for the main text of the report to be around 3000 words. There is no penalty for exceeding the word limit but it is strongly recommended to stay within these limits as you have a finite time in which to complete this report.

LEARNING OUTCOMES AND MARKING CRITERIA

These objectives satisfy all the learning outcomes of the module as re-stated below:

LO1: Demonstrate a thorough understanding of the main issues related to network computing

LO2: Critically evaluate requirements and problems that arise when advanced network computing applications are designed and implemented

LO3: Demonstrate practical ability in implementing an advanced network computing application

It does this by giving you the opportunity to understand how to apply the main issues of network computing to a practical application and to critically evaluate those through a report. For this assignment (Task 1) the LOs are marked together as follows:

Task 1: Threshold Coursework – The coursework specification will be distributed on the date shown above with a submission deadline also noted above. The assessment will require you to design and implement a network computing application as described above. There will be structured opportunities to get advice on the assessment between it being set and you submitting during the workshops.

The assessment of Task 1, which will be undertaken in Term 1, will be confirmed by the Panel of Examiners and, if necessary, the Board of Examiners will offer a reassessment of Task 1 in Term 2, subject to the re-assessment limitations of SR2.

A student who fails to achieve grade D⁻ in Task 1 at both the first and second attempt will not be eligible for any further assessment/re-assessment in the module.

Learning outcomes for the assessment	Assessment and marking criteria
There is no evidence that the threshold requirements could be met by the work presented.	F grade



There is some evidence that the threshold requirements are close to being met, however, fall short in attaining the expected standard, e.g. failure to create a working network computing application and no clear explanation of how the application works.	E grade
Threshold (D): The student has built and demonstrated a working network computing application that satisfies the functionality described in the assignment and has produced an adequate report.	D- grade

SUBMISSION INSTRUCTIONS

You must submit your coursework as a PDF file on Wiseflow by the deadline at 11am. You can follow the link to Wiseflow through the module's section on Brightspace or login in directly at <https://uk.wiseflow.net/brunel>. The name of your file should follow the normal convention set out in the student handbook, and must therefore include your student ID number (e.g., 0612345.pdf). It can also include the module code (e.g., CS2001_0612345.pdf).

AVOIDING ACADEMIC MISCONDUCT

Before working on and then submitting your coursework, please ensure that you understand the meaning of [plagiarism, collusion](#), and cheating (including [contract cheating](#)) and the seriousness of these offences. Academic misconduct is serious and being found guilty of it results in penalties that can reduce the class of your degree and may lead to you being expelled from the University. Information on what constitutes academic misconduct and the potential consequences for students can be found in [Senate Regulation 6](#).

You may also find it useful to read this [PowerPoint presentation](#) which explains, in plain English, the different kinds of misconduct, how to avoid (even accidentally) committing them, how we detect misconduct, and the common reasons that students give for engaging in such activities.

If you are experiencing difficulties with any part of your studies, remember there is always help available:

- Speak to your personal tutor. If you're not sure who your tutor is, please ask the Taught Programmes Office (TPOcomputerscience@brunel.ac.uk).
- Alternatively, if you prefer to speak to someone outside of the Department you can contact the [Student Support and Welfare](#) team.

EXPECTATIONS OF ARTIFICIAL INTELLIGENCE USE

The University has general guidance on [using artificial intelligence in your studies](#).

LATE COURSEWORK

The clear expectation is that you will submit your coursework by the submission deadline stated in the study guide. In line with the University's policy on the late submission of coursework (revised in July 2016), coursework submitted up to 48 hours late will be accepted but capped at a threshold pass (D- for undergraduate or C- for postgraduate). Work submitted over 48 hours after the stated deadline will automatically be given a fail grade (F).

Please refer to the [Computer Science student information pages](#) and the [Coursework Submission Procedure](#) pages for information on submitting late work, penalties applied and procedures in the case of Extenuating circumstances.

