COMP1549 (2023/2024)	**Advanced Programming Coursework**	**Contribution 100% of course**
Course Leader: Dr Muhammad Taimoor Khan	**Faculty Header ID:**	**Deadline Date March 29th, 2024**

Coursework Submission Requirements

- **An electronic copy of your work for this coursework must be fully uploaded by midnight on or before the Deadline Date.**
- **For this coursework you must submit a single ZIP file.**
- **Make sure that any files you upload are virus-free and NOT protected by a password or corrupted otherwise they will be treated as null submissions.**
- **Your work will not be printed in colour. Please ensure that any pages with colour are acceptable when printed in Black and White.**

Coursework Regulations

1. **If you have Extenuating Circumstances, you may submit your coursework up to two weeks after the published deadline without penalty, but this is subject to acceptance of your claim by the Faculty Extenuating Circumstances Panel.**
2. **Late submissions will be dealt with in accordance with University Regulations.**
3. **Coursework submitted more than two weeks late may be given feedback but will be recorded as a non-submission regardless of any extenuating circumstances.**
4. **Do not ask the lecturers for extensions to published deadlines - they are not authorised to award an extension.**
5. **The coursework must be submitted as above. Under no circumstances can they be accepted by academic staff.**

Please refer to the University Portal for further detail regarding the University Academic Regulations concerning Extenuating Circumstances claims.

DETAILED SPECIFICATION

This coursework contributes to 100% of your final grade of this module. This coursework can be submitted **either individually or in a group of up to 5 students**. The coursework will include the submission and evaluation of the following two elements:

1) project source code [**70 marks**]

2) project report [**30 marks**]

The coursework must be implemented in Java and should be submitted as

1. **a single ZIP** file that contains "**src**" folder of project implementation, a **PDF** of the project report, and a "**contributions.txt**" file that includes the percentage of contribution of each member of the group. In principle, we expect each member to contribute equally to the coursework where the sum of contribution from all members of a group equals 100%.

2. and a **short video** of 5-10 minutes demonstrating key requirements of the course work task as stated in the section *Assessment Details* on the next page.

The assessment of the coursework report will be based on its completeness, correctness, readability, and conformance to the expected format. While the assessment of the source code will be based on the evaluation and testing of your code against the detailed technical tasks listed in the next section of the document.

Coursework Task

The aim of the coursework is to implement a Graphical User Interface (GUI) or Command-Line Interface (CLI) based networked distributed system for a group-based client-server communication, which conforms with the following requirements. When a new member joins (connects), he/she may provide the following parameters as an input (i.e., command-line parameters or parameters set via the GUI):

1. an ID (please ensure that each member is assigned a unique ID),
2. a port and IP address of the server, and

If a member is the first connection in the group, the member automatically becomes the coordinator and must be informed about it at the start-up. Later, whoever joins must be informed about the details of the current coordinator. The coordinator maintains state of active group members by checking periodically (i.e., say every 20 seconds). Any member can request details of existing members from the coordinator and will receive everyone's IDs, IP addresses and ports including the current group coordinator. Based on the details, everyone should be able to send private or broadcast messages to every other member through the server. Anyone can leave the group at any time, however, if the coordinator leaves, then any new member will become a coordinator. In case any member leaves, the communication among the remaining group members should not be interrupted. The system should print out messages sent to/by the members.

Importantly, your implementation can either run automatically or manually. In the former case, your program may simulate the above task automatically without accepting any input parameters from users while running. For instance, a client and server may exchange different messages periodically.

In the latter case, your program requires user input to simulate the task. For instance, a user will type a message from a specific client to send it to other members or server. Any member can quit by a simple ctrl-C command (CLI) or having a Quit button (GUI).

For further details about the technical background required for the task, please see lecture material, or ask the instructors.

The project implementation **must demonstrate the following programming principles and practices, which contribute directly to the final grade**:

- Using design patterns
- JUnit based testing of the application
- Fault tolerance
- Use of Version Control System (VCS)

Assessment guidelines

Report and Implementation: The assessment of the project report is based on the overall technical quality, relevancy and completeness, contributions, and the details in supporting your solution and implementation. While the assessment of the project implementation is purely based on the correct implementation of technical requirements (as described in Coursework Task) and its adherence to various principles (as stated in Coursework Task).

Demonstration video: Specifically, the video demonstration should be prepared adhering to the following description. Each group must submit ONLY one demonstration video. The maximum duration of the video demonstration should be 10 minutes. Any ONE group member can record the video. Please submit your video through Moodle/Panopto at the latest by **April 5, 2024, 23:30 UK time**. Please ensure that the name of your demonstration file is as follows **STUDENT_EMAIL_ID_OF_THE_RECORDER.xyz**. A specific submission area for coursework demonstration will be made available on Moodle/Panopto. In principle, you can use any tool you wish to record your video. However, here are two different tutorials that show how you can use Panopto to record a video, and the other shows how you can use PowerPoint. However, you can use any tool of your choice.

Your video should demonstrate the following two items:

A. **Scenario Demonstration**
   1) You should run a server and three clients, where one of them is coordinator
   2) Demonstrate how the coordinator works
   3) Send a message from one of the clients and reply to it back (show private messaging and broadcast messaging).
   4) Quit one client (NOT coordinator) and show the rest two can still communicate
   5) Run another client
   6) Quit coordinator now and demonstrate that new coordinator is automatically selected

B. **Implementation Inspection**
   1) Highlight the code in your implementation that implements different design patterns.
   2) Highlight the code in your implementation that implements different components.

3) Highlight the code in your implementation that implements different JUnit tests.

Make sure that you demonstrate each feature as outlined in the coursework specification. You should specifically state what you are demonstrating - e.g. "**Now I will show design patterns**". You should also mention any additional requirements you have implemented or highlight aspects that you are proud of.

Please ensure that you show all the above-mentioned scenario and inspection parts but bear in mind that your video should be no longer than 10 minutes, so you shouldn't spend too much time on each part.

Here are a few tips on how you can reduce the length of the video:

1. Run the client and servers before you start the recording
2. Do a practice demonstration before you start recording
3. If you find that you are spending time waiting for some operation, etc. then you can always pause the recording

For project implementation, breakdown of the marks is as follows:

- The project should demonstrate the following programming principles and practices:
  - Group formation, connection, and communication [10 marks]
    - A group should be correctly formed connecting with all members where all members can communicate without any error.
  - Group state maintenance [10 marks]
    - The state of the group must be maintained correctly. This includes recording of the messages exchanged among members of the group with timestamps.
  - Coordinator selection [10 marks]
    - A correct implementation to automatically choose the coordinator even when the existing coordinator is disrupted/disconnected abnormally.
  - Use of design patterns [10 marks]
    - Adequate use of various design patterns in the implementation of the project.
  - Fault tolerance [10 marks]
    - Adequate strategy implementation for the fault tolerance, when a member terminates abnormally or when a coordinator terminates abnormally.
  - JUnit based testing of the application [10 marks]
    - Desired testing for implementation of all of the main requirements.
  - Use of Azure DevOps (Version Control System) [10 marks]
    - Setup of a Git repository for the code on Azure DevOps and regular commits to the repository (at least once per week)

You must include all group members in the Azure DevOps project to ensure that you can work collaboratively on the code and, when the marking schedule is published, you must also include your marker and ensure that they have Basic permissions to view the repository.

For the project report, you are asked to adhere to ALL the following rules:

- The report should be written using Times New Roman font size 11.
- The paper length should not exceed **5 pages double-column** according to the template (loosely based on the IEEE paper format), which can be seen on the next page. This is an upper limit to give you the flexibility to write an appropriate paper.

- The table of content of the report should include the following sections:
    - Introduction [04 marks]
        - This section should be a brief explanation to what the coursework is about, what you did for the coursework, and how your report is organized. You must also include a link to your Git repository on Azure DevOps.
    - Design/Implementation [10 marks]
        - This section should explain how you implemented each technical requirement. You should also brief the environment did you used for the implementation. You may use visuals (e.g., UML) to describe the design of implementation. Importantly, you should justify the implementation choices for key contributions, e.g., design patterns, and unit tests.
    - Analysis and Critical Discussion [08 marks]
        - This section should explain the results for running the code of all technical tasks. You should also explain how you achieved modularity, fault tolerance, testing and through which components. Importantly, you should state limitations or weaknesses of your implementation choices.
    - Conclusions [04 marks]
        - The conclusion based on analysis and implementation.
    - Presentation style [04 marks]
        - The presentation includes structure and contents of the report. The contents of the report should be adequate supported by reasonable justification.

Note: There is a 20% penalty if the paper does not abide by all above rules. Furthermore, first the submission will be evaluated based on the above criteria. Later, marks for each group member will be calculated based on their submitted percentage of contribution. For instance, if a group of 5 students with equal contribution (i.e., 20% each) achieves 80%, then a member of the group who has 20% contribution will get full 80% but if someone had 10% contribution, he would get only 40% score.

Assessment criteria

The assessment criteria for the technical task implementation and report are provided in a separate spreadsheet named “Rubrik.xlsx”.

Title

Author 1 name and author 2 name
COMP1549: Advanced Programming
University of Greenwich
Old Royal Naval College
United Kingdom

Abstract—**This is a very summary of the work produced, including a brief explanation of the topic and key results. This is the first piece of text that the reader will see. So, make sure it is well-written. A good length is one or two paragraphs.**

It worth 5% of the report.

Key words: *(keyword 1, keyword 2, keyword 3;, keyword 4; choose four relevant keywords or terms that characterize the topic of your choice)*

I. Introduction

Here, you describe the motivation and rationale behind the specific topic. What makes the area interesting or important, where is it used in the real world, and why does it make sense to measure the performance metrics that you have chosen. Use this space to set the scene for the rest of your paper.

Note that it is important to follow this template from the beginning to the end. Do not change fonts (Times New Roman 11), sizes or anything else.

The introduction worth 10% of the report.

Note that the whole paper needs to be precisely 5 pages. If you exceed the 5-page limit you will lose marks.

II. Design/implementation

Also known as "design" of your implementation, this is among the most important aspects of the paper. Here, you need to explain what salient features of your solution design and implementation are.

The related work section worth 40% of the report.

III. Analysis and Critical Discussion

This section will be the core part of the report, which will state all the analysis and arguments of the investigated aspect.

The analysis and discussion section worth 30% of the report.

IV. Conclusions

Here, you briefly summarise the work carried out and suggest possible future work. The conclusions section is like the abstract with the addition of the future work suggestion or perhaps more detail in the summarization of the results of the previous section.

The Conclusions section worth 10% of the report.

Acknowledgement

This is an optional section, where, if you want, you can thank colleagues or family that helped you or supported you in relation to this paper.

References

[List and number all references used in this paper following the Harvard Referencing style. It is not terribly important whether in each reference you place the issue number, page, publisher etc. What matters is that you are consistent, and you use precisely the same format in all your references.

The references worth 5% of the report.

An example list of references following the Harvard referencing style is shown below. Note the appropriate use of italics:]

Young, H.D., Freedman, R.A., Sandin, T. and Ford, A. (2000) *Sears and Zemansky's university physics*. 10th edn. San Francisco: Addison-Wesley.

Bell, J. (2005) *Doing your research project*. 4th edn. Maidenhead: Open University Press.

Jackson, G. (2000) 'Ports 1700-1840', in Clark, P. (ed.) *Cambridge urban history of Britain: Vol. 2 1540-1840*. Cambridge: Cambridge University Press, pp. 705-730.