Ain Shams University
Faculty of Engineering

Course Assessment Specification (CAS)

Programme Title : Computer Engineering and Software Systems
Coursework Title : Programming Project
Module Name (UEL) : Computer Networks
Course Name (ASU) : Computer Networks
Module/Course Code : EG7533 / CSE351
Level UEL/ASU : 5 / 3
UEL Credit Rating : 15 Credits **ASU Credit Rating : 3 Credits**
Weighting : 35%
Maximum mark available:

- 35 on software projects (1 project)

Lecturer : Prof. Ayman M. Bahaa-Eldin, Dr. Karim Ahmed Awad El Sayed Emara.

Contact : If you have any issues with this coursework, you may contact your lecturer.

Contact details are: Email: ayman.bahaa@eng.asu.edu.eg, karim.emara@cis.asu.edu.eg

Hand-out Date : As shown in submission matrix
Hand-in Date : As shown in submission matrix
Hand-in Method : Submission through LMS
Feedback Date : Your work will be marked and returned within two weeks.

Introduction

This coursework is itemized into several parts to get the 35 marks associated to it.
You must use the templates provided by the instructor to prepare your work.
All assignments and projects will be handed-in electronically, while quizzes and exams are written
.

Learning Outcome to be assessed

2. Explain the use of socket programming in networking applications
4. Recognize the differences between computational and communication overheads
5. Evaluate network performance
10. Work and communicate effectively in a team

Detail of the task

Attached separate file for each task.

89% and above:

Your work must be of outstanding quality and fully meet the requirements of the coursework specification and learning outcomes stated. You must show independent thinking and apply this to your work showing originality and consideration of key issues. There must be evidence of wider reading on the subject. In addition, your proposed solution should:

- illustrate a professional ability of drafting construction details,
- express a deep understanding of the in-hand problem definition,
- and applying, masterly, the learned knowledge in the proposed solution.

76% - 89%:

Your work must be of good quality and meet the requirements of the coursework specification and learning outcomes stated. You must demonstrate some originality in your work and show this by applying new learning to the key issues of the coursework. There must be evidence of wider reading on the subject. In addition, your proposed solution should:

- illustrate a Good ability of drafting construction details,
- express a very Good understanding of the in-hand problem definition,
- and applying most of the learned knowledge, correctly, in the proposed solution.

67% - 76%:

Your work must be comprehensive and meet all of the requirements stated by the coursework specification and learning outcomes. You must show a good understanding of the key concepts and be able to apply them to solve the problem set by the coursework. There must be enough depth to your work to provide evidence of wider reading. In addition, your proposed solution should:

- illustrate a moderate ability of drafting construction details,
- express a good understanding of the in-hand problem definition,
- and applying most of the learned knowledge, correctly, in the proposed solution.

60% - 67%:

Your work must be of a standard that meets the requirements stated by the coursework specification and learning outcomes. You must show a reasonable level of understanding of the key concepts and principles and you must have applied this knowledge to the coursework problem. There should be some evidence of wider reading. In addition, your proposed solution should:

- illustrate a fair ability of drafting construction details,
- express a fair understanding of the in-hand problem definition,
- and applying some of the learned knowledge, correctly, in the proposed solution.

Below 60%:

Your work is of poor quality and does not meet the requirements stated by the coursework specification and learning outcomes. There is a lack of understanding of key concepts and knowledge and no evidence of wider reading. In addition, your proposed solution would be:

- Illustrate an inability of drafting construction details,
- Failed to define the parameters, limitations, and offerings of the in-hand problem,
- Failed to apply correctly the learned knowledge for proposing a valid solution.

Academic Misconduct

The University defines Academic Misconduct as 'any case of deliberate, premeditated cheating, collusion, plagiarism or falsification of information, in an attempt to deceive and gain an unfair advantage in assessment'. This includes attempting to gain marks as part of a team without making a contribution. The department takes Academic Misconduct very seriously and any suspected cases will be investigated through the University's standard policy. If you are found guilty, you may be expelled from the University with no award.

It is your responsibility to ensure that you understand what constitutes Academic Misconduct and to ensure that you do not break the rules. If you are unclear about what is required, please ask.

1. **This is a group (3-4 students) project,**
2. **Groups are announced on the LMS**
3. **Only one member of the group is required to submit.**
4. **Delivery will be on 4 Phases as described here and timed on the LMS**

Project Title: Implementing a server agent of a standard protocol

Introduction:

The goal of this project is to analyze and implement a server agent for DHCP, DNS, or HTTP/2 protocols that is compliant with the RFC. The project should work seamlessly with common off-the-shelf clients such as a typical browser or Windows DHCP/DNS client agents. The project will be divided into four phases, each building on the previous one, to demonstrate compliance details with the RFC. The project should follow behaviors, scenarios, messages format and sequence, and produce RFC-compliant outcomes or error codes.

Backlog of Requirements and User Stories are according to the RFC.

RFCs can be found here https://datatracker.ietf.org/

- **HTTP/2 (HTTP/2.0)**

RFC 7540: Hypertext Transfer Protocol Version 2 (HTTP/2). This RFC specifies HTTP/2, a major revision of the HTTP network protocol, designed to improve performance over its predecessor HTTP/1.1. HTTP/2 introduces features like multiplexing of requests, header compression, and server push. It aims to reduce latency and improve the overall web browsing experience while maintaining compatibility with HTTP/1.1 semantics.

RFC 7541: HPACK: Header Compression for HTTP/2. This document defines HPACK, the header compression algorithm used by HTTP/2. It focuses on reducing the size of headers and improving performance in situations where large headers need to be transferred.

- **Dynamic Host Configuration Protocol (DHCP)**

RFC 2131: Dynamic Host Configuration Protocol (DHCP) defines DHCP, a protocol used to automate the assignment of IP addresses, subnet masks, gateways, and other network parameters. It allows devices on a network to request configuration information from a DHCP server dynamically.

RFC 2132: DHCP Options and BOOTP Vendor Extensions. This RFC details additional options that can be used with DHCP, extending the base protocol to support various configuration parameters such as subnet masks, routers, and domain names.

- **Domain Name System (DNS)**

RFC 1034: Domain Names – Concepts and Facilities. This foundational RFC defines the concept of the Domain Name System (DNS), which is responsible for translating human-readable domain names into IP addresses. It outlines the basic structure and operation of DNS.

RFC 1035: Domain Names – Implementation and Specification. This RFC provides detailed specifications for the DNS protocol, including message formats, query types, and response structures. It focuses on how the DNS protocol operates in practice.

RFC 2181: Clarifications to the DNS Specification. This RFC clarifies ambiguities and updates certain aspects of the original DNS specification (RFC 1034 and RFC 1035) to address issues encountered during deployment.

Project Phases

Phase 1: Project Planning and Design

1. **Objective:** Define the project scope, goals, and functionalities. Create a detailed design document outlining the architecture, components, and communication protocols.
2. **Deliverables:**
    - Project proposal outlining objectives and scope.
    - Design document detailing system architecture, components, and communication protocols.

    Steps
    - Define user authentication mechanisms.
    - Outline the overall project scope, including key functionalities.
    - Design the system architecture, specifying components and their interactions.
    - Demonstrate communication/message protocols according to the RFC.

Phase 2: Basic Client-Server Setup

1. **Objective:** Implement the foundational elements of the system, including basic client-server communication using Python and sockets.
2. **Deliverables:**
    - Basic server application.
    - Establish a connection for basic RFC features.

Steps

- **I**mplement a basic server application capable of handling multiple client connections.
- Establish a TCP connection for user authentication (if required).
- Implement a simple command-line interface showing the server status.

Phase 3: Extend Server features

1. **Objective:** extend the developed server agent with RFC features and error codes.
2. **Deliverables:**
    - Modified server to support features detailed in RFC.
    - Demonstrate functionalities with different use cases.

Phase 4: Server Optimization

1. **Objective:** Enhance the developed agent by optimizing communication protocols for better performance according to the RFC.
2. **Deliverables:**
    - Optimize TCP and UDP usage based on different scenarios.
    - Conduct performance testing and document results.
    - Finalize the user interface and overall user experience.

    Steps

    - Optimize communication protocols based on different scenarios (e.g., TCP for reliability, UDP for real-time interactions).
    - Conduct performance testing to evaluate the scalability of the system.
    - Finalize the user interface, considering usability and user experience.

Final Project Presentation and Documentation

1. **Objective:** Compile and present the completed project, emphasizing the learned concepts and demonstrating the implemented features.
2. **Deliverables:**
    - Finalized codebase.
    - User documentation detailing installation, configuration, and usage instructions.
    - Presentation slides and video showcasing the application and discussing the key design decisions.