

**Arab Academy for Science and Technology and Maritime Transport  
Computer Science Curriculum  
Course Syllabus**

<b>Course Code:</b> CS432	<b>Course Title:</b> Network Protocols & Programming	<b>Classification:</b> E	<b>Coordinator's Name:</b> Prof. Dr. Aliaa Youssif	<b>Credit Hours:</b> 3
------------------------------	---	-----------------------------	---	---------------------------

<b>Pre-requisites:</b> <ul style="list-style-type: none"> <li>• CE231 (Introduction to Networks)</li> <li>• CS244 (Advanced Programming Applications)</li> </ul>	<b>Co-requisites:</b> None	<b>Schedule:</b> Lecture: 2 hours Tutorial-Lab: 2 hours		
--	-------------------------------	---	--	--

**Office Hours:**

**Course Description:**

The course is an introduction to network programming using the application programming interface known as sockets. In addition, several design alternatives for client/server applications will be presented along with tradeoffs. Upon successful completion of the course, the student should be able to develop network-aware applications that involve unicast and multicast communications from the grounds up. Hands-on experience will be gained through several programming assignments, which

require knowledge of the C and/or Java programming language.

**Textbook:**

W. Richard Stevens, Bill Fenner, and Andrew Rudoff, *Unix Network Programming, The Sockets Networking API*, Volume 1, 3<sup>rd</sup> Edition, Addison-Wesley.

**References:**

- Douglas E. Comer and David L. Stevens, *Internetworking with TCP/IP Volume III: Client-Server Programming and Applications*, Linux/POSIX sockets version, Prentice Hall, 2001.
- Douglas E. Comer, *Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture*, 5<sup>th</sup> Edition, Prentice Hall.

Course Objective/Course Learning Outcome:

Contribution to Program Student Outcomes:

1. Understand network protocols operation details and design methodology.

(SO1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

2. Apply programming of unicast and multicast network applications.

Differentiate client/server design alternatives along with tradeoffs.

(SO6) Apply computer science theory and software development fundamentals to produce computing-based solutions.

**Course Outline:**

- |   |  |
|---|--|
| <ol style="list-style-type: none"><li>1. Introduction to TCP/IP</li><li>2. The Network Layer: IP Suite</li><li>3. The Transport Layer: TCP and UDP</li><li>4. Elementary Sockets</li><li>5. Elementary TCP Sockets</li><li>6. TCP Client/Server Example</li><li>7. I/O Multiplexing</li></ol> | <ol style="list-style-type: none"><li>8. Socket Options</li><li>9. Elementary UDP Sockets</li><li>10. Name and Address Conversions</li><li>11. Multicasting</li><li>12. Client/Server Design Alternatives</li><li>13. Presentation of projects</li></ol> |
|---|--|

**Grade Distribution:**

7th Week Assessment (30%)

12th Week Assessment (20%)

Year Work (10%)

Final Exam (40%)

Policies:

Attendance:

AASTMT Education and Study Regulations (available at [aast.edu](http://aast.edu))

Academic Honesty:

AASTMT Education and Study Regulations (available at [aast.edu](http://aast.edu))

Late Submission:

*Late submissions are graded out of 75% (1 week late), 50% (2 weeks late), 25% (3 weeks late), 0% (more than 3 weeks late)*