

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

<b>Course Code:</b> CS321	<b>Course Title:</b> Systems Programming	<b>Classification:</b>	<b>Coordinator:</b> Dr. Mohamed Mostafa <b>Lecturer:</b>	<b>Credit Hours:</b> 3
<b>Pre-requisites:</b> <ul style="list-style-type: none"> <li>• CE243 (Introduction to Computer Architecture)</li> <li>• CS243 (Object-Oriented Programming)</li> </ul>	<b>Co-requisites:</b> None	<b>Schedule:</b> Lecture: 2 hours Tutorial-Lab: 2 hours		
<b>Office Hours: (Room 255)</b> <b>Thursday 8:30 a.m. -10:30 p.m.</b>				
<b>Course Description:</b> This course introduces the concepts and techniques of system programming. The course covers the following topics: Machine Architecture; Machine Language; Assembly Language; Two Pass Assemblers; One Pass Assemblers; Macro Facilities; Conditional Macros; Macro Processors; Loaders; Linkers; Formal Languages; The Design of Translators (Compilers and Interpreters).				
<b>Textbook:</b> Chattopad Hyad. Santanu, <i>System Software</i> , Prentice Hall of India.				
<b>References:</b> <ol style="list-style-type: none"> <li>1. John J. Donovan, <i>Systems Programming</i>, Central Book Co.</li> <li>2. David A. Watt, <i>Programming Language Processors</i>, Prentice Hall.</li> <li>3. Leland L. Beck, <i>System Software: An Introduction to Systems Programming</i>, Addison Wesley, Longman Inc.</li> </ol>				

Course Objective/Course Learning Outcome:	Contribution to Program Student Outcomes:
1. Understand the architecture of a hypothetical machine, its assembly language, and macro language.	(SO-1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Write programs in assembly language.	
3. Describe the syntax of different programming languages using formal grammar.	
4. Understand different object file formats.	
5. Master the concepts and theory behind the implementation of high level programming languages.	
6. Design assemblers, linkers and loaders and assess related design decisions.	(SO-3) Communicate effectively in a variety of professional contexts.
<p><b>Course Outline:</b></p> <p>Week 1. Introduction to Systems Programming  Week 2. Simplified Instructional Computer (SIC)  Week 3. Sic and Sic/xe programming examples  Week 4. Simple two pass assembler  Week 5. Object code generation  Week 6. Machine dependent assembler features  Week 7. 7th Week Exam  Week 8. Machine Independent Assembler features ( literals , symbols, expression)</p>	<p>Week 9. One-pass assembler  Week 10. Loader  Week 11. Linker  Week 12. 12th Week Exam  Week 13. Macros  Week 14. Introduction to Compiler  Week 15. Compiler design stages  Week 16. Final Exam</p>
<p><b>Grade Distribution:</b></p> <p>7th Week Assessment (30%):  Exam (20%) + Section Quiz 5% + Implementing a SIC machine two pass Assembler 5%</p> <p>12th Week Assessment (20%):  Exam (15%) + upgrading 7<sup>th</sup> week project to SIC/XE two pass Assembler 5%</p>	

Year Work (10%):

Adding to the project the SIC/XE two pass Assembler features (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)*