GAVILAN COLLEGE

5055 Santa Teresa Blvd
Gilroy, CA 95023

Course Outline

COURSE: CSIS 12L DIVISION: 50 ALSO LISTED AS:

TERM EFFECTIVE: Spring 2022 CURRICULUM APPROVAL DATE: 03/08/2022

SHORT TITLE: ASSEMBLY LANG LAB

LONG TITLE: Assembly Language Programming Lab

Units	Number of Weeks	Type	Contact Hours/Week	Total Contact Hours
1	18	Lecture:	0	0
		Lab:	3	54
		Other:	0	0
		Total:	3	54

COURSE DESCRIPTION:

Supplemental practice in coursework associated with this course is provided. Concurrent enrollment in CSIS 12 is required. This is a pass/no pass course. COREQUISITE: CSIS 12 Assembly Language Programming

PREREQUISITES:

COREQUISITES:
CSIS 12

CREDIT STATUS: D - Credit - Degree Applicable

GRADING MODES
P - Pass/No Pass

REPEATABILITY: N - Course may not be repeated

SCHEDULE TYPES:
04 - Laboratory/Studio/Activity
04A - Laboratory - LEH 0.65
05 - Hybrid
71 - Dist. Ed Internet Simultaneous
73 - Dist. Ed Internet Delayed LAB
73A - Dist. Ed Internet LAB-LEH 0.65

STUDENT LEARNING OUTCOMES:

By the end of this course, a student should:

1. Design, code, document, analyze, debug and test introductory level assembly programs.

COURSE OBJECTIVES:

By the end of this course, a student should:

1. Write assembly programs.
2. Work in binary and hexadecimal number bases.
3. Write simple macros.
4. Trace and debug at the assembly level.

CONTENT, STUDENT PERFORMANCE OBJECTIVES, OUT-OF-CLASS ASSIGNMENTS

Curriculum Approval Date: 03/08/2022

6 Hours

Lab Content: Running basic assembly language programs. Examine the number system and how it's used in the computer system. Create, save and print programs. Learn how to use standard input and output. Exercises: Use computer or calculators to perform binary to hex, hex to binary number conversion. Relate the number system to how the machine interprets them. Explain the relationship between high-level language and assembly language. Examine how memory maps are used in a computer system. Provide an overview of the DOS operating system. Do assignments that involve number systems, and the structure of machine codes. Create, modify, execute, debug, and print a simple assembly language program.

6 Hours

Lab Content: Create a basic program to do number conversions. Research operating system memory structure. Research operating system interrupt structure. Research operating system IO structure. Exercises: Study accumulator modeling. Study how registers are modeled in an 8-bit environment. Study how registers are modeled in a 16-bit environment. Study how registers are modeled in a 32-bit environment. Create, modify, execute, debug, and print an assembly language program that uses accumulators, registers and hexadecimal numbers. Explain the difference between real mode memory addressing and protected mode memory addressing and demonstrate how to implement them in a computer system.

6 Hours

Lab Content: Write assembly language programs using opcode MOV. Exercises: Study and review immediate addressing, direct addressing, register indirect addressing, base-plus-index address, register relative addressing, base relative-plus-index addressing and scaled index addressing. Explain the operation of each program memory-addressing mode and select the appropriate one to accomplish certain tasks. Describe the sequence of events that place data onto the stack or remove data from the stack.

6 Hours

Lab Content: Write a program that involves opcode MOVSX, MOVZX. Write a program that involves opcode PUSH, POP. Write a program that involves opcode BSWAP, XCHG, XLAT. Write a program that involves opcode IN, OUT. Write a program that involves opcode LEA, LDS, LES, LFS, LGS, LSS. Write a program that involves opcode HAFH, SAHF. Write a program that involves opcode MOVS, LODS, STORS, INS, OUTS. Exercises: Study the move functions. Study the push, pop functions. Study the load address functions. Study the string functions. Explain the operation of each data movement instruction with applicable addressing modes. State the purposes of the assembly language pseudo-operations and key words. Select the appropriate assembly language instruction to accomplish a specific date movement task.

6 Hours

Lab Content: Write assembly language programs to perform add, subtract, multiplication, division. Write assembly language programs to perform negation and comparison. Write assembly language programs to perform increment and decrement.

Exercises: Study opcodes which can perform add, subtract, multiply, divide and other possible arithmetic functions. Study opcodes, which can perform negation and comparison functions. Study opcodes, which can perform as counters for increment and decrements. Use arithmetic and logic instructions to accomplish simple binary, BCD, and ASII arithmetic. Create, modify, execute, debug, and print an assembly language program that uses three types of loops.

6 Hours

Lab Content: Write assembly language programs involving logic instructions AND, OR, Exclusive-OR, NOT. Write assembly language programs involving shifts, rotates, and logical compare (TEST). Exercises: Study logical instructions AND, OR Exclusive-OR, NOT. Study how to use assembly language programs to perform shifts, rotates and logical compare.

6 Hours

Lab Content: Write assembly language programs using opcode: JA, JAE, JB, JBE, JC. Write assembly language programs using opcode: JG, JGE, JL, JLE. Write assembly language programs using opcode: JNC, JNE, JNO, JNS, JNP. Write assembly language programs using opcode: JO, JP, JS, JCXZ, JECX. Use both conditional and unconditional jump instructions to control flow of a program. Use the relational assembly language statements: IF, REPEAT, WHILE. Use the call and return instructions to include procedures in the program structure. Exercises: Study jump instructions involving conditional jump. Study conditional jump and conditional sets. Study jump if overflow is set. Study jump if the register is zero.

10 Hours

Lab Content: Write programs using MASM assembler and linker program. Write programs using EXTRN and PUBLIC to set up library files. Write and use MACRO and ENDM to develop macro sequences. Exercises: Study MASM assembler and linker programs. Study EXTRN and PUBLIC functions.

2 Hours

Lab Final Exam.

METHODS OF INSTRUCTION:

Discussion, Demonstration, Guided Practice

METHODS OF EVALUATION:

Problem-solving assignments

Evaluation Percent 50

Evaluation Description

40% - 70% Lab Exercises, Exams

Skill demonstrations

Evaluation Percent 40

Evaluation Description

30% - 60% Computer Demonstration Exercises

Objective examinations

Evaluation Percent 10

Evaluation Description

0% - 20% Multiple Choice, True/False, Matching Items, Completion

REPRESENTATIVE TEXTBOOKS:

Assembly Language for x86 Processors (8th Edition), Irvine, Kip, Prentice-Hall (Pearson Learning), 2020.

ISBN: 978-0135381656

Rationale: This edition is an interactive electronic textbook which can be used for lab.

12+ Grade Verified by: Publisher

ARTICULATION and CERTIFICATE INFORMATION

Associate Degree:

CSU GE:

IGETC:

CSU TRANSFER:

Transferable CSU, effective 200770

UC TRANSFER:

Transferable UC, effective 200770

SUPPLEMENTAL DATA:

Basic Skills: N

Classification: Y

Noncredit Category: Y

Cooperative Education:

Program Status: 1 Program Applicable

Special Class Status: N

CAN:

CAN Sequence:

CSU Crosswalk Course Department:

CSU Crosswalk Course Number:

Prior to College Level: Y

Non Credit Enhanced Funding: N

Funding Agency Code: Y

In-Service: N

Occupational Course: C

Maximum Hours:

Minimum Hours:

Course Control Number: CCC000382204

Sports/Physical Education Course: N

Taxonomy of Program: 070710