GAVILAN COLLEGE
5055 Santa Teresa Blvd
Gilroy, CA 95023

Course Outline

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

TERM EFFECTIVE: Fall 2016 CURRICULUM APPROVAL DATE: 11/23/2015

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
05 - Hybrid
72 - Dist. Ed Internet Delayed

STUDENT LEARNING OUTCOMES:
1. Create, modify, execute, debug, and print a simple assembly language program.

Measure: Homework, exercises, quizzes.
PLO: 2
ILO: 7, 3, 2
GE-LO:
Year assessed or anticipated year of assessment: 2016

2. Create, modify, execute, debug, and print an assembly language program that uses three types of loops.
Measure: Homework, exercises, quizzes.
PLO: 2, 4
ILO: 3, 7, 2
GE-LO:
Year assessed or anticipated year of assessment: 2016

3. Create, modify, execute, debug, and print an assembly language program that uses accumulators, registers and hexadecimal numbers.
Measure: Homework lab exercises, projects
PLO: 2, 4
ILO: 3, 7, 2
GE-LO:
Year assessed or anticipated year of assessment: 2016

4. Create, modify, execute, debug, and print an assembly language program that uses decision and jump statements.
Measure: Exercises, homework, quizzes
PLO: 2, 3, 4
ILO: 3, 7, 2
GE-LO:
Year assessed or anticipated year of assessment:

5. Create, modify, execute, debug, and print an assembly language program that uses five different arithmetic operations and four arithmetic functions.
Measure: Homework, quizzes, projects.
PLO: 2, 4, 3
ILO: 7, 3, 2
GE-LO:
Year assessed or anticipated year of assessment:

PROGRAM LEARNING OUTCOMES:

1) Student will code, debug, document, test, and run complex C++ programs.

2) Student will write programs in at least three different programming languages, and compare and contrast the philosophies and comparative advantages of each these languages.

3) Students will demonstrate professional conduct by meeting project deadlines, and participating in self-managed teams.

4) Student will create algorithms to solve programming problems, and implement those algorithms.

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

Curriculum Approval Date: 11/23/2015
WEEK HOURS CONTENT
1-2 Lab Lab:
6 Running basic assembly language programs.
Create, save and print programs.
Learn how to use standard input and output.
Homework:
Use computer or calculators to perform binary to hex,
hex to binary number conversion.
Relate number system to how machine interprets them
Do assignments that involve number systems, and
structure of machine codes.
Read chapter in the textbook related to number systems,
assembly languages.
3-4 Lab Lab:
6 Create basic program to do number conversions.
Research operating system memory structure.
Research operating system interrupt structure.
Research operating system IO structure.
Homework:
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.
5-6 Lab Lab:
6 write assembly language programs using opcode MOV.
Homework:
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.
7-8 Lab Lab:
6 Write program involve opcode MOVSX, MOVZX
Write program involve opcode PUSH, POP
Write program involve opcode BSWAP, XCHG, XLAT
Write program involve opcode IN, OUT
Write program involve opcode LEA, LDS, LES, LFS, LGS,
LSS.
Write program involve opcode HAFH, SAHF,
Write program involve opcode MOVS, LODS, STORS, INS, OUTS
Homework:
Study the move functions
Study the push, pop functions
Study the load address functions
Study the string functions
9-10 Lab Lab:
6 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.
Homework:
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.
11-12 Lab Lab:
6 Write assembly language programs involving logic instructions AND, OR, Exclusive-OR, NOT.
Write assembly language programs involving shifts, rotates, and logical compare (TEST).
Homework:
Study logical instructions AND, OR Exclusive-OR, NOT
Study how to use assembly language programs to perform shifts, rotates and logical compare.
13-14 Lab Lab:
6 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
Homework:
Study jump instructions involving conditional jump.
Study conditional jump and conditional sets.
Study jump if overflow set.
Study jump if register is zero.
15-16 Lab Lab:
6
17-18 Lab Lab:
6 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.
Homework:
Study MASM assembler and linker programs.
Study EXTRN and PUBLIC functions.
Read chapter in the textbook related to programming with C++.
STUDENT PERFORMANCE OBJECTIVES:
Weeks 1-2

Students should have a thorough understanding of the number system and how it's used in the computer system.
Students should learn the relationship between, high-level language and assembly language.
Students should learn how memory maps are used in a computer system.
Overview of the DOS operating system.
Weeks 3-4
Students should understand how to model accumulator, base index register, counter and data register in 8 bit, 16 bit and 32 bit environment. How to use the registers to accomplish memory addressing? Students should understand the difference between real mode memory addressing, protected mode memory addressing and how to implement them in a computer system.
Weeks 5-7
Student should learn the operation of each data-addressing mode.
Students should be able to 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.
Weeks 8-10
Students should be able to explain the operation of each data movement instruction with applicable addressing modes.
Understand the purposes of the assembly language pseudo-operations and key words.
Select the appropriate assembly language instruction to accomplish a specific date movement task.
Weeks 11-12
Students should be able to use arithmetic and logic instructions to accomplish simple binary, BCD, and ASII arithmetic.
Weeks 13-14
Use AND, OR and Exclusive-AND, to accomplish binary bit manipulation.
Use the shift and rotate instruction.
Weeks 15-16
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 program structure.
Weeks 17-18
Write assembler code using macros
Able to link and include libraries
ASSIGNMENTS:
Included in the content section of the course outline.

METHODS OF INSTRUCTION:
Lecture, computer demonstration, projects, lab exercises.

METHODS OF EVALUATION:

The types of writing assignments required:
Written homework
Reading reports
Lab reports
The problem-solving assignments required:
Homework problems
Field work
Exams
The types of skill demonstrations required:
Class performance
Performance exams
The types of objective examinations used in the course:
Multiple choice
True/false
Matching items
Completion
Other category:
None
The basis for assigning students grades in the course:
Writing assignments: 10% - 40%
Problem-solving demonstrations: 30% - 50%
Skill demonstrations: 10% - 50%
Objective examinations: 5% - 20%
Other methods of evaluation: 0% - 0%

REPRESENTATIVE TEXTBOOKS:

Required:
Irvine, Kip. Assembly Language for x86 Processors (7th Edition). Pearson, 2014. Or other appropriate college level text.
Reading level of text, Grade: 12+ Verified by: ev

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: CSIS
CSU Crosswalk Course Number: 12L
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