GAVILAN COLLEGE
5055 Santa Teresa Blvd
Gilroy, CA 95023

Course Outline

COURSE: CSIS 18 DIVISION: 50 ALSO LISTED AS:

TERM EFFECTIVE: Spring 2019 CURRICULUM APPROVAL DATE: 05/14/2018

SHORT TITLE: UNIX/C++ PROG

LONG TITLE: UNIX/C++ Programming

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

COURSE DESCRIPTION:

An introduction to the C++ programming language and the UNIX operating system. Topics include programming on a UNIX system, including C/C++ language, shell programming, and the interface between C++ and UNIX. This course has the option of a letter grade or pass/no pass. Concurrent enrollment in CSIS 18L is required. COREQUISITE: CSIS 18L UNIX/C++ Programming Lab ADVISORY: CSIS 48 UNIX Operating System, CSIS 10 BASIC Programming or other programming experience.

PREREQUISITES:

COREQUISITES:
CSIS 18L

CREDIT STATUS: D - Credit - Degree Applicable

GRADING MODES
L - Standard Letter Grade
P - Pass/No Pass

REPEATABILITY: N - Course may not be repeated

SCHEDULE TYPES:
02 - Lecture and/or discussion
05 - Hybrid
72 - Dist. Ed Internet Delayed

STUDENT LEARNING OUTCOMES:

1. Create programs using calculations and selection.
Measure of assessment: Homework, projects, lab exercises.
2. Create programs using loops and arrays.
Measure of assessment: Homework, projects.
3. Create programs using methods and objects.
Measure of assessment: Homework, lab exercises, projects.
4. Use UNIX editors, UNIX file commands, and UNIX utilities to create and manipulate program and data files.
Measure of assessment: Homework, projects, tests, quizzes.

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

Curriculum Approval Date: 05/14/2018
WEEK HOURS CONTENT:
1-2 6 Lecture: History of UNIX and C/C++. Using emacs
or vi editor. File commands, directories, and printing.
File commands, directories, and printing. Creating,
compiling, executing, and printing programs.
Introduction to programming concepts. Structural
programming. Simple variables: types, declaring,
initializing, and comments.
Types: integers, floats,
and chars. Declaring and using constants. Arithmetic
operators and mathematical error: precision problems
and overflow.
Homework: The students learn basic UNIX commands
Necessary for programming. The students prepare,
compile, and execute the sample program. The students
modify the sample program and hand in the
modifications. The students write
programs using
several types of variables and arithmetic operators.
3-4 6 Lecture: Control Structures. Selection, if, if/else
statements. While and for repetition structures.
Do/While structures. Switch, break and continue
statements. Relational and logical operators.
UNIX
programming debugging tools: debuggers, lint. UNIX
debugging tools: debuggers, lint, tracers, and
profilers.
Homework: Use several types of decision statements.
Use relational and logical operations. Use simple,
compound, and nested if
statement. Use debugging tools
available for C/C++ under UNIX.
5-6 6 Lecture: Math functions. Simple functions, defining
functions, calling functions, prototype functions.
C vs. C++ function syntax. Local and global variables,
scope of variables. Storage classes,

mathematical
recursion. Returning values from functions. UNIX file management tools: make, archive, tar. Function templates
and overloading.
Homework: Prepare programs with functional and use math library functions. Prepare programs that have local and global variables that demonstrate scope of rules. Prepare programs that solve mathematical problems using recursion. Prepare
programs that pass
values into functions and that return values. Prepare overloaded functions and function templates.
7-8 6
Lecture: Mid-term test. Simple arrays: 1-D arrays. Character string arrays. Type and uses of subscripts. Multiple dimension arrays. Processing arrays
with loops
and nested loops. Arrays and functions. UNIX shell programs: variables, decisions, and loops.
Homework: Prepare
programs that use numeric and string
arrays. Use several different types of subscripts. Prepare programs that use loops to process the arrays. Write UNIX shell programs with variables, decisions, and loops.
9-10 6 Lecture: Pointer variable declaration and initialization. Address of variables, vs. value of variable. Pointer expressions and pointer arithmetic. Using pointers as a function
arguments. Arrays as
pointers.
Homework: Prepare programs that declare and use pointers. Prepare programs that use pointers as function arguments. Prepare programs that use pointer expressions and pointer arithmetic.
11-12 6 Lecture: Structures. Defining structures and initializing structures. Using structure members. Arrays of structures and
structures as function
arguments. Limitations of structures. Introduction to classes, and classes vs. structures. Defining, initializing, and using structures.
Homework: Prepare programs with structures. Prepare programs with arrays of structures. Prepare
programs
that use simple classes.
13-14 6 Lecture: More on classes. Class scope and accessing

class members
and functions. Constructors and
destructors. Const objects and const member functions.
Friend functions and friend classes. Static class
members. Operators overloading. Overloading unary
and binary operators.
Homework: Prepare programs that use different
types
of class scope and member access. Prepare programs that
use overloaded constructors. Prepare programs that use
friend functions
and static class members.
15-16 6 Lecture: Inheritance and polymorphism. Base classes
and derived classes. C++ stream input/output. Stream
manipulators and format states. Templates.
Homework: Write programs that use Inheritance and
polymorphism. Write programs that use base
classes
to create derived classes. Write programs that use
stream input/output and use manipulators and
formatting.
17-18 6 Lecture: Throwing and catching exceptions. Other
error-handling techniques. File processing. Creating
and accessing a sequential file.
The preprocessor.
Final Exam.
Lab: Write programs that have errors that can be
thrown and caught. Write programs that create and use
sequential files. Use the preprocessor to include
libraries and files.
COURSE OBJECTIVES:
Weeks 1-2
The students use enough UNIX to get a program
running.
The students print the program and output so they can hand it in.
The students use the different types of simple variables.
The students use
simple arithmetic operators with the variables.
The students use comments to document their programs.
Weeks 3-4
The students write programs that use if/then and if/then/else
statements.
The students write programs using several types
of loop structures.
The students write programs with nested if statements.
The students write programs that have multiple relational and
logical operators.
The students use UNIX/C debugging tools.

Weeks 5-6
The students write programs that uses math
functions and their
functions.
The students write programs that uses local and global variables.
The students
write programs using recursion techniques to solve
mathematical problems.
The students write programs that pass different types of
arguments.
The students write programs that overload functions.
Weeks 7-8
The students write programs
that uses 1-D arrays with characters
and numbers.
The students write programs that uses multiple dimension arrays.
The students write programs that uses a loop to process arrays.
The students write programs that pass arrays to functions.
The students write UNIX shell programs with
variables, decisions,
and loops.
Weeks 9-10
The students
write programs that uses pointer variables.
The students write programs that uses pointers that need to be
initialized and compared.
The students write programs that use pointers as function arguments.
The students write programs that use arrays of pointers.
Weeks 11-12
The students write programs that use structures.
The students write programs that use an array of structures.
The students write programs that use classes.
Weeks 13-14
The students write programs that use complex classes and member
functions.
The students write programs that use public and private variables
and functions.
The students write programs that use static
class members and
operator overloading.
Weeks 15-16
The students
write programs that use inheritance and polymorphism.
The students write programs that use C++ stream input/output
processes.
The students write programs that use stream manipulators and format
states.
The students write programs that use templates.

Weeks 17-18
The students write programs that throw and catch errors.
The students write programs that create and use
sequential files.

METHODS OF INSTRUCTION:
Lecture, computer demonstration.
OUT OF CLASS ASSIGNMENTS:
Required Outside Hours: 108
Assignment Description: Reading textbook and posted class material.
Required Outside Hours:
Assignment Description: Work on sample programs, homework programs, and projects.

METHODS OF EVALUATION:
Writing assignments
Percent of total grade: 5.00 %
Writing assignments: 5% - 20% Written homework
Problem-solving assignments
Percent of total grade: 40.00 %
Problem-solving demonstrations: 40% - 60% Homework problems Quizzes Exams
Skill demonstrations
Percent of total grade: 15.00 %
Skill demonstrations: 15% - 50% Class performance Performance exams
Objective examinations
Percent of total grade: 10.00 %
Objective examinations: 10% - 40% Multiple choice True/false Matching items Completion

REPRESENTATIVE TEXTBOOKS:
Required Representative Textbooks
Walter Savitch. Problem Solving with C++ (10th Edition) . Pearson,2017.
ISBN: 978-0134448282
Reading Level of Text, Grade: 12+ Verified by: MS Word

ARTICULATION and CERTIFICATE INFORMATION
Associate Degree:
CSU GE:
IGETC:
CSU TRANSFER:
Transferable CSU, effective 200630
UC TRANSFER:
Transferable UC, effective 200630

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: 18
Prior to College Level: Y
Non Credit Enhanced Funding: N
Funding Agency Code: Y
In-Service: N
Occupational Course: D
Maximum Hours:
Minimum Hours:
Course Control Number: CCC000227174
Sports/Physical Education Course: N
Taxonomy of Program: 070710