FAIRMONT STATE UNIVERSITY™

Office of the Provost
1201 Locust Avenue • Fairmont, West Virginia 26554
Phone: (304) 367-4101 • Fax: (304) 367-4902
www.fairmontstate.edu

MEMORANDUM

TO: Faculty Senate

FROM: Dr. Susan Ross

DATE: April 7, 2021

SUBJECT: Curriculum Proposal #20-21-28

The intent of this proposal is to propose minor revisions of COMP 1130 – Principles of Programming II.

cc: Richard Stephens
Lori Schoonmaker
Stephanie Gabor
Laura Ransom
Mahmood Hossain

Proposal Number:	#20-21-28
School/Department/Program:	Science and Technology / Computer Science and Math
Preparer/Contact Person:	Dr. Mahmood Hossain
Telephone Extension:	4967
Date Originally Submitted:	March 3, 2021
Revision:	
Implementation Date Requested	Fall 2021

I PROPOSAL

The intent of this proposal is to propose minor revisions of COMP 1130 – *Principles of Programming II*.

II DESCRIPTION OF THE PROPOSAL

A. Deletion of course(s) or credit(s) from program(s) Not Applicable

B. Addition of course(s) or credit(s) to program(s) Not Applicable

C. Provision for interchangeable use of course(s) with program(s) Not Applicable

D. Revision of course content COMP 1130 (Appendix A).

E. Other changes to existing courses Not Applicable

F. Creation of new courses Not Applicable

G. Attach an itemized summary of the present program(s) affected

The following majors/minors require COMP 1130:

BS in Computer Science, BS in Computer Science – Cybersecurity Concentration
BS in Electronics Engineering Technology (Elective)
BS in Mechanical Engineering Technology (Elective)
Minor in Computer Science
Minor in Game Design
Minor in Automation and Robotics

III. RATIONALE FOR THE PROPOSAL

1. Quantitative Assessment:

COMP 1130 is the second programming course for the Computer Science majors. This course is a continuation of the programming concepts covered in COMP 1120. Programming projects are assigned in C++. The object-oriented concepts part will be removed since those are covered in COMP 2200 and there is an overlap.

2. Qualitative Assessment:

IV. PROPOSAL AFFECTING OTHER COLLEGES/SCHOOLS Not Applicable

V. PROPOSAL AFFECTING GENERAL STUDIES Not Applicable

VI. ADDITIONAL COMMENTS Not Applicable

APPENDIX A

COMP 1130 Principles of Programming II

Current Description

COMP 1130 Principles of Programming II 4 hrs. Lecture/Lab Hours: The course consists of three hours of lecture and three hours of lab per week This course is a continuation of COMP 1120 and covers arrays, searching/sorting, pointers, classes, recursion, and advanced file I/O. The concept of object oriented programming is introduced. Projects in C++ are assigned to provide students with experience implementing multi-part applications using these concepts. PR: A grade of C or better in COMP 1120.

Revised Description

COMP 1130 Principles of Programming II 4 hrs. Lecture/Lab Hours: The course consists of three hours of lecture and three hours of lab per week This course covers arrays, searching/sorting, pointers, strings, recursion, and advanced file I/O. Programming projects are assigned in C++ to provide experience in implementing multi-part applications using these concepts. PR: A grade of C or better in COMP 1120.

Outcome	**Direct Assessment**	**Satisfactory Performance Standard**
Store and process data using one-dimensional and two-dimensional arrays.	Programming Project	A class average of 70% or more.
Compare/contrast basic searching and sorting techniques and apply them in code.	Quiz/Exam	A class average of 70% or more.
Access and process data using pointers.	Quiz/Exam	A class average of 70% or more.
Store and process text data using strings.	Programming Project	A class average of 70% or more.
Write recursive functions to solve simple recursively solvable problems.	Programming Project	A class average of 70% or more.
Read/write data using advanced file-processing techniques.	Quiz/Exam	A class average of 70% or more.
Develop programs using a uniform coding style and adequate documentation.	Programming Projects	A class average of 70% or more.

APPENDIX A

Detailed course outline:

- Overview of C++
  - Data Types
  - Arithmetic Operators
  - Input/Output
  - Decision Structures
  - Loop Structures
  - Functions
- Arrays
  - Declaring and Initializing an Array
  - Operations on Arrays
  - Two-Dimensional Arrays
  - Arrays as Function Parameters
  - Searching
  - Sorting
- Pointers
  - Declaring, Initializing, and Dereferencing a Pointer
  - Operations on Pointers
  - Dynamic Memory Allocation
  - Memory Leaks and Dangling Pointers
  - Pointers as Function Parameters
- Strings
  - C-type Strings
    - Declaration
    - Operations
  - C++ Strings
    - Declaration
    - Operations
  - Conversion between C-type and C++ Strings
- Recursion
  - Base Case and General Case
  - Avoiding Infinite Recursion
  - Examples
- Advanced File I/O
  - Reading Data from a Binary File
  - Writing Data into a Binary File
  - Random Access Files