Republic of the Philippines
SULTAN KUDARAT STATE UNIVERSITY
Isulan, Sultan Kudarat
College of Computer Studies
SY 2022 - 2023

UNIVERSITY VISION

A trailblazer in arts, science, and technology in the region.

UNIVERSITY MISSION

The University shall primarily provide advanced instruction and professional training in science and technology, agriculture, fisheries, education, and other related field of study. It shall undertake research and extension services and provide progressive leadership in its area of specialization.

UNIVERSITY GOAL

To produce graduates with excellence and dignity in arts, science, and technology.

UNIVERSITY OBJECTIVES

a. Enhance competency development, commitment, professionalism, unity, and true spirit of service for public accountability, transparency, and delivery of quality services.
b. Provide relevant programs and professional training that will respond to the development needs of the region.
c. Strengthen local and international collaborations and partnerships for borderless programs.
d. Develop a research culture among faculty and students.
e. Develop and promote environmentally sound and market-driven knowledge and technologies at par with international standards.
f. Promote research-based information and technologies for sustainable development.
g. Enhance resource generation and mobilization to sustain the financial viability of the university.

Program Objectives and its Relationship to University Objectives:

PROGRAM OBJECTIVES (PO)	UNIVERSITY OBJECTIVES						
A graduate of BS in Information Technology can:	a	b	c	d	E	f	g
a. innovate technological concepts and ideas underpinning desired IT solutions;	/	/	/	/	/	/	/
b. administer competently the computer networks, systems development, software applications, hardware, and maintenance;	/	/	/	/	/	/	/
c. design industry-based applications, infrastructures, and technologies that will promote the advancement and development of the community;	/	/	/	/	/	/	/
d. Adopt various national and international industries standards in the practice of the profession; and	/	/	/	/	/	/	/
e. demonstrate professionalism in the social, environmental, and legal aspects of information technology.	/	/	/	/	/	/	/

1. Course Code	: CC112
2. Course Title	: Programming 1
3. Prerequisite	: NONE
4. Credits	: 3 UNITS

5. Course Description:

The course covers the use of general-purpose programming language to solve problems. The emphasis is to train students to design, implement, test, and debug programs intended to solve computing problems using fundamental programming constructs.

6. Course Learning Outcomes and Relationships to Program Objectives

Course Learning Outcomes	Program Objectives				
At the end of the semester, the students can:	a	b	c	d	e
a. analyze and design strategies for solving basic programming problems;	/	/			
b. know how to use variables and named constants;	/	/			
c. design a program with decision-making capabilities using conditional statements;	/	/	/		
d. develop simple programs that utilize looping statements;	/	/	/		
e. know how to use functions and procedures;	/	/	/		
f. use arrays to store, process, and sort data;	/	/	/		
g. manipulate string values of the variables	/	/	/		
h. open, write, and read sequential access file	/	/	/		
i. appreciate the importance of computer programming;	/	/	/	/	/
j. take responsibility for the proper use of computer and computer programs; and	/	/	/	/	/
k. manifest creativity, love, and respect for others.					/

7. Course Content

Course Objectives, Topics, Time Allotment	Desired Student Learning Outcomes	Outcomes-Based Assessment (OBA) Activities	Evidence of Outcomes	Course Learning Outcomes	Program Objectives	Values Integration
TOPIC 1: SKSU VMGO, CLASSROOM POLICIES, COURSE OVERVIEW, COURSE REQUIREMENTS, GRADING SYSTEM (1 hour)						
1. Discuss the VMGO of the university, classroom, and computer laboratory policies, scope of the course, course requirements, and grading system	1.1 Students can be aware of and appreciate the university's VMGO, classroom, and computer laboratory policies,	Individual participation by way of asking for clarification on VMGO, classroom policies and requirements, and	Teacher-student interaction	k	e	Value of appreciation

	course overview, requirements, and grading system.	grading system if deemed necessary				
TOPIC 2: INTRODUCTION TO PROGRAMMING (2 hours)						
2.1. History of Programming Languages	The student can: 2.1. learn the evolution of programming languages	Lectures and class discussions	Assignments	a, j, k	a, b, c, d, e	Value of listening and appreciation
2.2. Control Structures a. Sequence b. Selection c. Repetition	2.2. differentiate and describe the three (3) types of control structures					
TOPIC 3: CREATING COMPUTER SOLUTIONS TO PROBLEMS (3 hours)						
3.1. Programming Terminologies	The student can: 3.1. define the basic programming terminologies such as program, programmers, programming languages	Lectures and class discussions Formulating algorithm Flowcharting	Group discussions Bring home activities Assignments Quizzes	a, j, k	a, b, c, d, e	Value of listening and appreciation Value of logical analysis
3.2. Basic steps in creating program solutions to problems	3.2. analyze the problem, plan the algorithm, desk-check the algorithm, code the algorithm into a	Desk-checking				Value of unity and teamwork

3.3. Analysis tools 3.4. Desk-checking process	program, desk-check the program, and evaluate and modify the program 3.3. learn to use programming tools in analyzing a program such as IPO charts, algorithms, flowcharts, and pseudo-code 3.4. know how to desk-check the algorithm and computer program					
TOPIC 4. INTRODUCTION TO C++ PROGRAMMING (10 hours)						
4.1. C++ Basic Program 4.2. Preprocessor Directives	The student can: 4.1. create a basic C++ program 4.2. understand the meaning of preprocessor directives	Lectures and class discussions Problem-solving Writing a computer program	Graded Group Discussions Quiz Rubrics for Laboratory Exercises	a, b,	a, b	` Value of memory retention and familiarization

4.3. Header Files 4.4. Using Statements 4.5. Function Header and Body of the Main Program 4.6. COUT and CIN Statements 4.7. Insertion and Extraction Stream Operators 4.8. Comments Statements	4.3. identify the different header files in C++ 4.4. know how to utilize standard program statement 4.5. declare the main function and its body 4.6. know how to use cout and cin statements in a C++ program 4.7. understand the use of insertion and extraction stream operators 4.8. write the additional description of the program by using hashtag comment statements	Laboratory exercises				
TOPIC 5. VARIABLES, CONSTANTS AND ARITHMETIC OPERATORS (10 hours)						
	The student can:					

5.1. Variable Definition and declaration	5.1. define and declare variables in C++	Lectures and class discussions Class participation Laboratory activities Presentation of program source code and output	Quiz Graded Laboratory Activities	a, b	a, b	Value of exploration Value of actualization
5.2. Named Constant Variables	5.2. create a C++ program using constant variables					
5.3. Rules in naming Variables	5.3. familiarize the basic rules in naming simple variables and named constant variables					
5.4. Variable Data Types	5.4. learn the fundamental data types of variables and their memory requirements and allowable values					
5.5. Assigning Initial Value to Variables	5.5. learn how to put valid initial values to memory variables					
5.6. Types of Conversion	5.6. understand the implicit and explicit type conversions					
5.7. Assignment Statements	5.7. put value to a memory variable					

5.8. Arithmetic Operators 5.9. Getline() and Ignore() Functions 5.10. Formatting Floating-Point Numbers	using an assignment statement 5.8. familiarize standard arithmetic operators and their order of precedence 5.9. learn other built-in functions in the C++ program 5.10. know how to format floating-point numbers using the fixed, scientific, and setprecision() stream manipulators					
TOPIC 6. SELECTION STRUCTURE (10 hours)						
6.1. if and if/else statement 6.2. Comparison Operators	The student can: 6.1. create a C++ program using if and if/else statements 6.2. know how to use comparison operators in an if	Lectures and class discussions Class participation Group Activities Presentation of	Quiz Graded Laboratory Activities Graded score for question-and-answer activities	a, b, c	a, b, c	Value of exploration Value of logical analysis Unity and

	and if/else statements	program source code and output				teamwork
6.3. Logical Operator	6.3. use logical operator And(&&) and Or(||) in an if and if/else statements	Question and answer activities				
6.4. Nested Selection Structure	6.4. create a C++ program using the inner and outer if and if/else statements					
6.5. Switch Case Statement	6.5. convert multiple path selection structures or nested if/else statements to switch case form statements					
TOPIC 7. REPETITION STRUCTURE (10 hours)						
	The student can:					
7.1. While Loop Statements	7.1. create a C++ program using while loop statements	Lectures and class discussions Class participation	Graded Recitation . Graded Score for Laboratory Work Activity	a, b, c, d	a, b, c	Value of exploration Value of actualization
7.2. Do While Loop Statements	7.2. create a C++ program using do-	Group activities	Graded Hands-on			

	while loop statements	Question and answer activities	Exam			Unity and teamwork
7.3. For Loop Statements	7.3. create a C++ program using for loop statements	Hands-on activities				
7.4. Nested Loop	7.4. create a C++ program using nested looping statements					
TOPIC 8. FUNCTIONS (10 hours)						
	The students can:					
8.1. Definition of Function	8.1. understand the meaning of function in C++	Asynchronous discussions	Graded recitations	a, b, c, d, e	a, b, c	Value of actualization
8.2. Two Categories of a Function	8.2. differentiate a value-returning function and a void function	Interactive discussions Hands-on activities	Graded hands-on activities			Value of collaboration Value of logical analysis
8.3. Function Definition	8.3. identify the composition of a function definition					
8.4. Function Prototype	8.4. know how to declare a function prototype					
8.5. Function Declaration	8.5. create a C++					

and Calling a Function 8.6. Passing Variables to a Function	program that declares and calls a function 8.6. know how to pass variables to a function by value or by reference					
TOPIC 9. ARRAYS (12 hours)						
9.1. Array Variables 9.2. Defining Arrays 9.3. One-Dimensional Array 9.4. Passing a One-Dimensional Array to a Function 9.5. Two-Dimensional Array	The students can: 9.1. distinguish simple variables and array variables 9.2. define an array in C++ 9.3. declare, store, and display values to a one-dimensional array 9.4. know how to pass array variables to a function by value or by reference 9.5. declare, store, display values, and search data to a two-dimensional array	Asynchronous discussions Interactive discussions Hands-on activities Individual laboratory work Activity Presentation of program output	Graded question-and-answer activities Graded laboratory work activity Graded actual hands-on	a, b, c, d, e, f	a, b, c	Value of creativity Value of determination Value of actualization

9.6. Multi-Dimensional Arrays	9.6. write multi-dimensional array in a C++ program					
TOPIC 10. STRING MANIPULATION (12 hours)						
10.1. Manipulating Strings 10.2. Determining the Number of Characters Contained in a String 10.3. Removing Characters from a String 10.4. Accessing Characters Contained in a String 10.5. Replacing Characters in a String	The student can: 10.1. manipulate values of a string variable 10.2. determine the number of characters in a string of variables 10.3. remove one or more characters located anywhere in a string variable 10.4. access any number of characters in a string variable 10.5. replace a sequence of characters in a string variable	Asynchronous discussions Interactive discussions Actual hands-on activities Individual laboratory work Activity Presentation of program output Brainstorming	Graded class participation Graded laboratory works Graded hands-on activities Graded presentation of program output	a, b, c, d, e, f, g	a, b, c	Value of appreciation Value of exploration Value of collaboration Value of determination

10.6. Inserting Characters within a String 10.7. Searching a String 10.8. Concatenating Strings	with another sequence of characters 10.6. insert characters within a string 10.7. search a string variable to determine if it contains a sequence of characters 10.8. combining one string value with another string values					
TOPIC 11. SEQUENTIAL ACCESS FILES (10 hours)						
10.1. File Types 10.2. Using Sequential Access Files 10.3. Writing and Reading	The student can: 10.1. identify the different types of files such as sequential, random, and binary 10.2. create and open sequential access file 10.3. write and read	Asynchronous discussions Interactive discussions Research works Brainstorming Actual hands-on activities	Quizzes Graded class participation Graded laboratory works Graded hands-on activities Graded presentation	a, b, c, d, e, f, g, h, i, j, k	a, b, c, d, e	Value of appreciation Value of exploration Value of independency Value of creativity

Information to a Sequential Access File 10.4. Closing a Sequential Access File	data from a sequential access file 10.4. prevent a loss of data by closing an open sequential access file after the program uses it	Individual laboratory work Activity Presentation of program output	of program output			Value of output-oriented characteristic
Total number of hours with laboratory (94 hours)						

Lectures	36 hours
Laboratory	54 hours
Examination	4 hours

8. Course Evaluation

Course Requirements: 80% running program

Grading System:

MID-TERM and **FINAL-TERM**	
Participation/Attendance	5%
Quiz / Assignment	10%
Actual Hands-On Activities	35%
Exam	50%
TOTAL	**100%**

Schedule of Examination:

Midterm :
Final Term :
Classes End :

References:

Textbooks:

Diane Zak, An Introduction to Programming With C++ (5th Edition)
D.S Malik C++ Programming form Problem Analyis
Bjarne Stroustrup, The C++ Programming Language, 4th Edition
Steve Tale, C++: The Ultimate Beginners Guide to C++ Programing
Siddhartha Rao,C++ in One Hour a Day, Sams Teach Yourself (8th Edition)

Timothy Short, C++: Beginner to Pro Guide (C++ Programming 2016)
Nathan Clark,C++: Programming Basics for Absolute Beginners (Step-By-Step C++ Book 1)

Supplemental:

http://www.csulb.edu/colleges/coe/cecs/views/programs/undergrad/grade_prog.shtml
https://www.udemy.com/complete-c-programming-step-by-step-tutorial/
http://www.c4learn.com/c-programming/learn-c-programming-language-step-by-step-tutorials/

Prepared by:

GREGORIO C. ILAO, PhD
Faculty

Approved:

BENEDICT A. RABUT, DIT
Dean, College of Computer Studies