Republic of the Philippines
SULTAN KUDARAT STATE UNIVERSITY
College of Computer Studies
Isulan, Sultan Kudarat

CC113 – COMPUTER PROGRAMMING II

UNIVERSITY VISION

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

UNIVERSITY MISSION

The University shall primarily provide advance 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 trainings 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 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 to 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** : CC113
2. **Course Title** : Programming 2
3. **Prerequisite** : Programming 1
4. **Credits** : 3 UNITS
5. **Course Requirements** :

   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 fundamentals programming constructs.

6. **Course Learning Outcomes and Relationships to Program Objectives**

COURSE LEARNING OUTCOMES (CLOs)	PROGRAM OBJECTIVES (POs)				
At the end of the semester, the students can:	a	b	c	d	e
a. know on how to use functions and procedures;	/	/	/	/	/
b. use arrays to store, process and sort data;	/	/	/	/	/
c. manipulate string values of the variables;	/	/	/	/	/
d. create and develop programs using array, structures, functions and pointers	/	/	/	/	/
e. learn on how to use references and pointers;	/	/	/	/	/
f. appreciate the importance of computer programming;	/	/	/	/	/
g. take responsibility on the proper use of computer and computer programs; and	/	/	/	/	/
h. manifest creativity, love and respect for others.	/	/	/	/	/

7. Course Content

WEEK	CONTENT	INTENDED LEARNING OUTCOMES (ILO)	TEACHING AND LEARNING ACTIVITIES (TLA)	OUTCOMES – BASED ASSESSMENT (OBA)	COURSE LEARNING OUTCOMES (CLOS)
Week 1	Course Orientation SKSU VMGO, Classroom Policies, Course Overview, Course Requirements, Grading System	At the end of the week the students can: a. identify and discus the SKSU VMGO b. explain the relevance of the SKSU VMGO to their academic and professional development c. Summarize the key classroom policies d. List and explain the specific course requirements e. Demonstrate understanding of the importance of academic integrity and ethical conduct within the classroom f. Recognize the learning resources available for the course	● Presentation and Discussion ● Course Overview Analysis ● Open Forum	VMGO Reflection	g
Week 2 - 4	Review Lessons in Control Structures **ARRAY** ● One Dimensional ● Two Dimensional ● Multi-Dimensional Array	At the end of the week the student can: a. Explain the purpose and syntax of different control structures b. Apply control structures to solve programming problems c. Define and differentiate between one – dimensional,	● Lectures and Presentation ● Code Demonstration and Walkthroughs ● Hands on Coding exercises ● Group and Individual Problem-Solving Activities	● Coding Quizzes ● Programming Assignments ● Coding Exercises ● Debugging Exercises	b,d,f,g,h

		two – dimensional and multi – dimensional array d. Declare, initialize array of different dimensions using C++ programming language e. Access and manipulate elements within arrays using appropriate indexing. f. Design and Debug programs that solve real world problems using control structures and arrays g. Compare and contrast the efficiency of different control structures and array manipulation technique.	• Coding Challenge • Debugging Exercises • Code Analysis	• Project Presentation and Demonstration • Code Reviews	
Week 5 - 7	**FUNCTIONS** • Types of Functions in C++ • User Defined Functions • Passing Parameters to Functions • Local Variable • Global Variable • Static Variable • Register Variables • Functions and Arrays • Functions and Structures • Default Parameters • Inline Functions • Command Line Parameters • Function Overloading • Recursion	At the end of the weeks the student can: a. Define and explain the purpose of functions in C++ b. Differentiate between different types of functions c. Design and implement user – defined functions with appropriate return types and parameters d. Pass parameters to functions using different methods (pass – by – value, pass – by-reference) e. Explain the scope of lifetime of local, global, static and register variable	• Lectures and Presentation • Code Demonstration and Walkthroughs • Hands on Coding exercises • Group and Individual Problem-Solving Activities • Coding Challenge • Debugging Exercises • Code Analysis	• Coding Quizzes • Programming Assignments • Coding Exercises • Debugging Exercises • Project Presentation and Demonstration • Code Reviews	a,b,c,d,e,f,g,h

		f. Pass arrays and structures as parameters to functions g. Utilize default parameters in function definitions h. Implement inline functions for performance optimization i. Process command line parameters passed to a C++ program j. Implement function overloading to create functions k. Apply recursion to solve problems that can be broken down into smaller, self – similar subproblem l. Choose appropriate variable types and parameter passing methods for efficient and effective function design m. Debug and troubleshoot code that involves functions, variable scope, and parameter passing			
Week 8 - 9	**BUILT – IN FUNCTIONS** • Built in Functions • conio.h header file • stdio.h header file • math.h • type of functions / character functinons (ctype.h)	At the end of the week the student can: a. Identify and explain the purpose of common built – in functions b. Describe the purpose and functions provided by the conio.h, stdio.h, math.h, and ctype.h header files	• Lectures and Presentation • Code Demonstration and Walkthroughs • Hands on Coding exercises • Group and Individual Problem-Solving Activities	• Coding Quizzes • Programming Assignments • Coding Exercises • Debugging Exercises • Project Presentation and Demonstration	a,b,c,d,e,f,g,h

		c. Utilize the functions from stdio.h for input and output operations d. Apply functions from math.h for mathematical calculations e. Use functions from ctype.h for character classification and manipulation f. Explain the historical context and limitations in conio.h	• Coding Challenge • Debugging Exercises • Code Analysis	• Code Reviews	
Week 10 - 11	**STRUCTURES** • Declaring Structure • Array Structures • Nested Structure	At the end of the week the student can: a. Define and explain the purpose of structures b. Declare structures with appropriate data members (field) in C++. c. Initialize and access members of structure variables d. Create and manipulate array of structures. e. Implement nested structures to represent complex data relationships f. Apply structures to solve programming problems g. Trace the memory allocation of structures, array of structures, and nested structures.	• Lectures and Presentation • Code Demonstration and Walkthroughs • Hands on Coding exercises • Group and Individual Problem-Solving Activities • Coding Challenge • Debugging Exercises • Code Analysis	• Coding Quizzes • Programming Assignments • Coding Exercises • Debugging Exercises • Project Presentation and Demonstration • Code Reviews	a,b,c,d,e,f,g,h

Week 12 - 13	**C++ POINTERS** • Memory and References • Pointers • Operations on Pointers • Pointers and Arrays • Pointers and Strings • Array of Pointers • Pointers and Functions • Pointers and Structures • Memory Management with Pointers	At the end of the week the student can: a. Explain the concept of memory addresses, references, and pointers b. Declare and initialize pointer variable c. Perform pointer arithmetic and other pointer operations d. Utilize pointers to access and manipulate array elements e. Use pointers to manipulate strings f. Create and manage arrays of pointers g. Pass pointers as function arguments and return pointers from functions h. Access and modify structure members using pointers.	• Lectures and Presentation • Code Demonstration and Walkthroughs • Hands on Coding exercises • Group and Individual Problem-Solving Activities • Coding Challenge • Debugging Exercises • Code Analysis	• Coding Quizzes • Programming Assignments • Coding Exercises • Debugging Exercises • Project Presentation and Demonstration • Code Reviews	a,b,c,d,e,f,g,h
Week 14 - 16	**C++ STRINGS** • String • String Input • Array of Strings • String Function (string.h)	At the end of the weeks the student can: a. Define and explain the concept of strings as arrays of characters b. Declare and initialize string variables. c. Use different methods to input strings from the user (ex. Scanf, gets, fgets, cin.getline) d. Create and manipulate arrays of strings e. Utilize functions from string.h	• Lectures and Presentation • Code Demonstration and Walkthroughs • Hands on Coding exercises • Group and Individual Problem-Solving Activities • Coding Challenge • Debugging Exercises	• Coding Quizzes • Programming Assignments • Coding Exercises • Debugging Exercises • Project Presentation and Demonstration	a,b,c,d,e,f,g,h

		header file for string operations (ex. Strlen, strcpy, strcat, ctrcmp) f. Implement custom string manipulation functions g. Compare and contrast different string input methods and their advantages /disadvantages h. Solve programming problems that involve string manipulation and processing.	• Code Analysis	• Code Reviews	

8. COURSE REQUIREMENTS POLICIES

COURSE REQUIREMENTS	The following are the student's requirements: **Assignment, Quizzes, and Participation** • Submit assignments and hands on activities on time, a two-point deduction will be incurred for each day of late submission • Attend and participate in quizzes • Actively engage in class discussion, coding exercises and activities **Online Development** • Utilize onlinegdb for all coding activities • Be prepared to share the code for review and collaboration **Coding Logbook** • Maintain a detailed log of all code, notes, and debugging processes • The logbook will be regularly reviewed and utilized during the checking of activities. **Major Exams** • Pass a written exam on C++ theory • Pass a hands-on programming Exam using onlinegdb or an offline C++ compiler
COURSE POLICIES	**Attendance** • Attend class regularly 15 minutes late considered as absent, and 10 consecutive absences is considered as dropped **Uniform** • Wear complete during as prescribed in Student Handbook, PE uniforms are not allowed during the class hours.

	Cheating and Plagiarism • Academic dishonesty (including plagiarism) results in disciplinary action (per Student Handbook) **Respect Each Other** • Treat everyone with mutual respect • Maintain an inclusive learning environment • Avoid bullying in the class

9. GRADING SYSTEM AND RUBRICS FOR GRADING

MIDTERM / FINAL TERM		FINAL GRADE
Quizzes/ Assignment/Participation	10%	Final Grade = (Midterm Grade + Final Grade)/2
Hands on Activities	40%	
Exams (Written + Hands On Exam)	50%	
TOTAL	100%	

RUBRICS: C++ HANDS ON ACTIVITY AND EXAM

CRITERIA	100% (FULL CREDIT)	0% (NO CREDIT)
Program Execution and Accuracy	The program compiles without errors, executes successfully for all test cases, and produces completely accurate output according to the assignment specifications.	The program fails to compile, crashes during execution, or produces incorrect output for any test case.

10. REFERENCES

Textbooks:

McMullen, K., Matthews, E., & Parsons, J. J. (2022). Reading from Programming with C++. Boston, USA: Cengage Learning Inc. .
Prinz, U. K.-P. (2021). A Complete Guide to Programming in C++. Jones and Bartlett Publishers, Inc.
Zak, D. (2010). An Introduction to Programming With C++ (5th Edition). Course Technology.
Malik, D. (2017). C++ Programming from Problem Analysis. Cengage Learning.
Stroustrup, B. (2013). The C++ Programming Language, 4th Edition. Addison-Wesley Professional.
Tale, S. (2016). C++: The Ultimate Beginners Guide to C++ Programing.
Miaces, S. (2024). C++ Programming 2025 Guide for Beginners: Master the Fundamentals of C++ Programming with Hands-On Exercises and Real-World Applications.
Siddhartha. (2017). C++ in One Hour a Day, Sams Teach Yourself 8th Edition. Indiana, USA: Pearson Education Inc.
Short, T. (2016). C++: Beginner to Pro Guide (C++ Programming 2016) . Createspace Independent Publishing Platform.
Clark, N. (2017). C++: Programming Basics for Absolute Beginners (Step-By-Step C++).

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/
https://www.w3schools.com/cpp/cpp_oop.asp
https://www.programiz.com/cpp-programming/multidimensional-arrays

Prepared:

MARY GRACE L. PEROCHO
Faculty

Reviewed:

CERILO B. RUBIN, MIT
BSIT, Program Chairman

Approved:

ELBRENTO. ANTONIO, DIT
Dean, College of Computer Studies