



## INTECH 221 – COMPUTER PROGRAMMING

### UNIVERSITY VISION

A leading University in advancing scholarly innovation, multi-cultural convergence, and responsive public service in a borderless Region.

### UNIVERSITY MISSION

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

### UNIVERSITY STRATEGIC GOALS

- a. Deliver quality service to stakeholders to address current and future needs in instruction, research, extension, and production
- b. Observe strict implementation of the laws as well as the policies and regulations of the University
- c. Acquire with urgency state-of-the-art resources for its service areas
- d. Bolster the relationship of the University with its local and international customers and partners
- e. Leverage the qualifications and competences in personnel action and staffing
- f. Evaluate the efficiency and responsiveness of the University systems and processes

### INSTITUTIONAL OUTCOMES (IO)

- 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 OUTCOMES (PO) COMMON TO ALL PROGRAMS AND ITS RELATIONSHIPS TO INSTITUTIONAL OUTCOMES

A graduate of Industrial Technology student can:	INSTITUTIONAL OUTCOMES (IO)						
	a	b	c	d	e	f	g
a. Assume professional, technical, managerial and leadership roles in industrial organizations with the desired competence in the fields of practiced such as Automotive, Architectural drafting, Civil, Electrical, Electronics, Food and allied discipline.	/	/		/	/	/	

b.	Innovative explicit and modern technologies in the advancement of economy, society, technology and environmental sustainability.	/	/		/	/	/
c.	Generate research-based information and technologies at par from international standard; and	/		/	/	/	
d.	Promote and transfer knowledge and technologies for effective and efficient school - industry partnership.	/	/	/	/	/	

1. COURSE CODE: INTECH 221

2. COURSE TITLE: Computer Programming

3. PREREQUISITE: NONE

4. UNITS: 3

#### 5. COURSE DESCRIPTION

The course provides the student with a sound background in the principles of fundamentals in procedural programming. The course helps the student to develop analytical thinking through understanding logic formulation using the techniques in flowcharting. It encompasses a structured programming language, its environment, and control structures. The course aids the student in solving simple to complex problems by developing programs using a structural programming language.

#### 6. COURSE LEARNING OUTCOMES (CLO) AND ITS RELATIONSHIPS TO PROGRAM OUTCOMES

COURSE LEARNING OUTCOMES (CLO)	PROGRAM OBJECTIVES (PO)			
	a	b	c	d
At the end of the course, a student can:				
a. analyze problems and formulate logical solutions through the use of flowcharting techniques.	/	/	/	
b. grasp core principles of procedural programming, including variables, data types, operators, and expressions.	/	/	/	
c. gain a deep understanding of structured programming principles, including control structures (sequence, selection, and iteration), modularity, and code organization.	/	/	/	
d. be proficient in using a specific C++ programming language and its development environment to write, compile, and execute programs.	/	/	/	
e. apply their knowledge to develop programs that solve both simple and complex problems, demonstrating their ability to translate logical solutions into working code.	/	/	/	/
f. learn basic debugging and testing strategies to identify and resolve errors in their programs	/	/	/	
g. develop good coding practices, including writing readable and maintainable code with appropriate documentation.	/	/	/	/

## 7. COURSE CONTENTS

WEEK	CONTENT	INTENDED LEARNING OUTCOMES (ILOs)	TEACHING & LEARNING ACTIVITIES (TLA)	OUTCOMES - BASED ASSESSMENT (OBA)	COURSE LEARNING OUTCOMES (CLO)
Week 1	<b>Course Orientation</b> SKSU VMGO, Classroom Policies, Course Overview, Course Requirements, Grading System	<p>At the end of the week the students can:</p> <ul style="list-style-type: none"> <li>a. identify and discuss the SKSU VMGO</li> <li>b. explain the relevance of the SKSU VMGO to their academic and professional development</li> <li>c. Summarize the key classroom policies</li> <li>d. List and explain the specific course requirements</li> <li>e. Demonstrate understanding of the importance of academic integrity and ethical conduct within the classroom</li> <li>f. Recognize the learning resources available for the course</li> </ul>	<ul style="list-style-type: none"> <li>• Presentation and Discussion</li> <li>• Course Overview Analysis</li> <li>• Open Forum</li> </ul>	VMGO Reflection	
Week 2	<b>INTRODUCTION TO PROGRAMMING</b> <ul style="list-style-type: none"> <li>• History of Programming languages</li> <li>• Basic Steps in Writing Creating Program</li> </ul>	<p>At the end of the week the student can:</p> <ul style="list-style-type: none"> <li>a. define programming and explain its role in problem – solving and technology</li> </ul>	Lecture with Interactive Q & A	Written Quiz	a, b, c

	b. describe the historical evolution of programming languages c. identify and explain the basic steps involved in creating a computer program d. Recognize and explain fundamental programming concepts (variables, data types, operators, control structures). e. Apply basic problem – solving techniques to develop simple algorithms for given problems	Group Discussion  Real – World Examples Demonstrations  Timeline Creation Activity  Research and Presentation on specific languages  Flowcharting exercises  Concept Mapping  Hands on exercises	Class Participation  Oral Examination  Flowcharting Assignments Concept Mapping  Algorithm Design Problems		
Week 3	<b>INTRODUCTION TO C++ PROGRAMMING</b> ✓ Structures of Program ✓ C++ Basic Program <ul style="list-style-type: none"><li>• Preprocessor Directives</li><li>• Header Files</li><li>• Program Statements</li><li>• Using cout</li><li>• Comments and comment syntax</li><li>• Forms and Syntax</li><li>• Library Functions</li><li>• Statements</li><li>• Functions Header and Body of the Main Program</li><li>• cin statements</li></ul>	At the end of the weeks the student can:  a. Apply basic problem – solving techniques to develop simple algorithms for given problems  b. Write simple C++ program that utilize basic program statements; including cout and cin for input / output operations Use comments to annotate C++ code,  c.	Lecture with code examples Code Walkthrough  Hands – on Coding Exercises  Live Coding demonstration  Problem Solving with simple C++ programs	Written Quiz  Code Structure Analysis  Programming Assignments  Syntax Error Identification	b, c d, e ,f,g

	<ul style="list-style-type: none"> <li>insertion and extraction stream operators</li> <li>comments statements</li> </ul>	<p>d. explaining their purpose and different comment syntax forms</p> <p>e. Recognize and apply correct C++ syntax, including the use of library functions from relevant header files</p> <p>e. Utilize insertion (&lt;&lt;) and extraction (&gt;&gt;) stream operators for effective input and output operations in C++ program</p>	<p>Code annotation exercises</p> <p>Syntax error analysis</p>		
Week 4 - 6	<b>VARIABLES</b> <ul style="list-style-type: none"> <li>Variables Definition and declarations</li> <li>different types of variables</li> <li>assignment statements</li> <li>rules in naming variables</li> <li>constant variables</li> <li>input with cin</li> <li>assigning initial values to variables</li> <li>types of conversion</li> <li>arithmetic, comparison and logical operators</li> <li>get line() ignore() functions</li> <li>formatting floating point numbers</li> </ul>	<p>At the end of the week the student can:</p> <p>a. define and declare variables of different data types (int, float, double, char, string) in C++</p> <p>b. Apply assignment statements to store values in variables and assign initial values during declaration</p> <p>c. Adhere to the rules for naming variables in C++ and understand the concept of constant variables.</p> <p>d. Perform implicit and explicit type conversions between different data</p>	<p>Lecture with code examples</p> <p>Variable declarations exercises</p> <p>Data Type Demonstration</p> <p>Hands – on coding exercises</p> <p>Variable naming quizzes</p> <p>Coding Exercises</p> <p>Operator Demonstration</p>	<p>Written Quiz</p> <p>Code Analysis</p> <p>Code Review</p> <p>Error Analysis</p> <p>Operator Expression Evaluation</p>	b, c, d, e, f, g

		e. types Utilize arithmetic, comparison, and logical operators to perform calculations and make decision in C++ programs	and coding with arithmetic, comparison and logical operators  Input handling exercises		
Week 7 - 9	<b>SELECTION STRUCTURE</b> ✓ If, else if, nested else if and switch statement	<p>At the end of the week the student can:</p> <ul style="list-style-type: none"> <li>a. explain the functionality and syntax of if, else if, and nested if statements in C++ and identify appropriate use cases</li> <li>b. implement if, else if, and nested if statements in C++ programs to make decisions based on different conditions</li> <li>c. explain the functionality and syntax of switch statements in C++ and identify the appropriate use cases</li> <li>d. implement switch statements in C++ programs to efficiently handle multiple – choice selection</li> <li>e. analyze programming problems and select the most appropriate selection structure (if,</li> </ul>	<p>Lecture with code examples  Flowchart demonstration  Conditional Logic exercises  Hands – on Coding exercises</p>	<p>Written Quiz  Code Analysis  Programming Assignments Code Review  Problem Analysis</p>	b, c, d, e, f, g

		else if, else if or switch) for efficient and readable code.			
Week 10 - 13	<b>REPETITION STRUCTURE</b> <ul style="list-style-type: none"> <li>✓ while loop</li> <li>✓ do – while loop</li> <li>✓ for loop</li> </ul>	<p>At the end of the week the student can:</p> <ol style="list-style-type: none"> <li>a. explain the functionality and syntax of while loops in C++ and identify appropriate use cases</li> <li>b. Implement while loops to perform repetitive tasks based on a condition</li> <li>c. Explain the functionality and syntax of do – while loops and identify appropriate use cases</li> <li>d. Implement do – while loops to perform repetitive tasks, ensuring the loop body executes at least once</li> <li>e. Explain the functionality and syntax of for loops and identify appropriate use cases</li> <li>f. Implement for loops to perform repetitive tasks with a known number of iterations</li> <li>g. Analyze programming problems and select the most appropriate repetition structure</li> </ol>	Lecture with code examples  Flowchart demonstration  Conditional Logic exercises  Hands – on Coding exercises	Written Quiz  Code Analysis  Programming Assignments Code Review  Problem Analysis	b, c, d, e, f, g

		(while, do – while, or for) for efficient and readable code			
Week 14 - 16	<b>ARRAYS</b> ✓ One – dimensional Array ✓ Two – dimensional Array	<p>At the end of the week the student can:</p> <ul style="list-style-type: none"> <li>a. Explain the concept of one – dimensional arrays, their declaration, initialization, and access in C++</li> <li>b. Implement one – dimensional arrays program to store and manipulate collections of data</li> <li>c. Explain the concept of two – dimensional arrays, their declaration, initialization and access</li> <li>d. Implement two – dimensional arrays to store and manipulate tabular data</li> <li>e. Perform common array operations such as searching, sorting, and manipulation of array elements</li> <li>f. Apply one – dimensional and two – dimensional arrays to solve real – world problems.</li> </ul>	<p>Lecture with code examples</p> <p>Algorithm demonstration</p> <p>Conditional Logic exercises</p> <p>Hands – on Coding exercises</p>	<p>Written Quiz</p> <p>Code Analysis</p> <p>Programming Assignments</p> <p>Code Review</p> <p>Problem Analysis</p>	b, c d, e, f, g

		(while, do – while, or for) for efficient and readable code			
WEEK 14 - 16	<b>ARRAYS</b>  ✓ One – dimensional Array ✓ Two – dimensional Array	<p>At the end of the week the student can:</p> <ul style="list-style-type: none"> <li>a. Explain the concept of one – dimensional arrays, their declaration, initialization, and access in C++</li> <li>b. Implement one – dimensional arrays program to store and manipulate collections of data</li> <li>c. Explain the concept of two – dimensional arrays, their declaration, initialization and access</li> <li>d. Implement two – dimensional arrays to store and manipulate tabular data</li> <li>e. Perform common array operations such as searching, sorting, and manipulation of array elements</li> <li>f. Apply one – dimensional and two – dimensional arrays to solve real – world problems.</li> </ul>	<p>Lecture with code examples</p> <p>Algorithm demonstration</p> <p>Conditional Logic exercises</p> <p>Hands – on Coding exercises</p>	<p>Written Quiz</p> <p>Code Analysis</p> <p>Programming Assignments</p> <p>Code Review</p> <p>Problem Analysis</p>	b, c, d, e, f, g

## 8. COURSE REQUIREMENT POLICIES

COURSE REQUIREMENTS	<p>The following are the student's requirements:</p> <p><b>Assignment, Quizzes, and Participation</b></p> <ul style="list-style-type: none"><li>• Submit assignments and hands on activities on time, a two-point deduction will be incurred for each day of late submission</li><li>• Attend and participate in quizzes</li><li>• Actively engage in class discussion, coding exercises and activities</li></ul> <p><b>Online Development</b></p> <ul style="list-style-type: none"><li>• Utilize onlinedb for all coding activities</li><li>• Be prepared to share the code for review and collaboration</li></ul> <p><b>Coding Logbook</b></p> <ul style="list-style-type: none"><li>• Maintain a detailed log of all code, notes, and debugging processes</li><li>• The logbook will be regularly reviewed and utilized during the checking of activities.</li></ul> <p><b>Major Exams</b></p> <ul style="list-style-type: none"><li>• Pass a written exam on C++ theory</li><li>• Pass a hands-on programming Exam using onlinedb or an offline C++ compiler</li></ul>
COURSE POLICIES	<p><b>Attendance</b></p> <ul style="list-style-type: none"><li>• Attend class regularly 15 minutes late considered as absent, and 10 consecutive absences is considered as dropped</li></ul> <p><b>Uniform</b></p> <ul style="list-style-type: none"><li>• Wear complete during as prescribed in Student Handbook, PE uniforms are not allowed during the class hours.</li></ul> <p><b>Cheating and Plagiarism</b></p> <ul style="list-style-type: none"><li>• Academic dishonesty (including plagiarism) results in disciplinary action (per Student Handbook)</li></ul> <p><b>Respect Each Other</b></p> <ul style="list-style-type: none"><li>• Treat everyone with mutual respect</li><li>• Maintain an inclusive learning environment</li><li>• Avoid bullying in the class</li></ul>

## 9. GRADING SYSTEM AND RUBRICS FOR GRADING

### MIDTERM / FINAL TERM

Quizzes/ Assignment/Participation

10%

### FINAL GRADE

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)
<b>Program Execution and Accuracy</b>	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++ Programming.
- 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++).

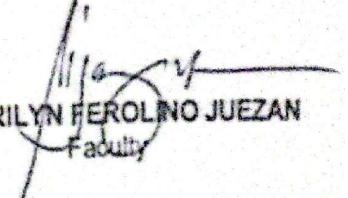
**Supplemental:**

[http://www.csulb.edu/colleges/coe/cecs/views/programs/undergrad/grade\\_prog.shtml](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.w3schools.com/cpp/cpp_oop.asp)  
<https://www.programiz.com/cpp-programming/multidimensional-arrays>

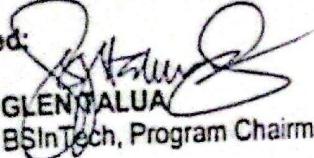
**Prepared:**

  
**MARY GRACE LAGGAMEN PEROCHO**  
Faculty

  
**CELIA ROSE JALON NOTA**  
Faculty

  
**MARILYN FEROLINO JUEZAN**  
Faculty

**Reviewed:**

  
**GLEN TALUA**  
BSInTech, Program Chairman

**Approved:**

  
**CHARLIE G. MAGHANOZ, EdD.**  
Dean, College of Industrial Technology

**SYLLABUS IN INTECH 221 (COMPUTER PROGRAMMING)**