VISAYAS STATE UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY
ICT Building, Visayas State University
Visca, Baybay City, Leyte, PHILIPPINES
Telefax: 053 565 0600 local 1022
Email: dcst@vsu.edu.ph
Website: www.vsu.edu.ph

OUTCOMES-BASED EDUCATION (OBE) COURSE SYLLABUS

CSci 21
Data Structures and Algorithms

I. UNIVERSITY INFORMATION

1. Vision of the University

   A globally competitive university for science, technology, and environmental conservation

2. Mission of the University

   Development of a highly competitive human resource, cutting-edge scientific knowledge and innovative technologies for sustainable communities and environment.

3. VSU Quality Policy Statement

The Visayas State University (VSU), a globally competitive university of science and technology and environmental conservation, is created by law to develop highly competitive human resource, cutting- edge scientific knowledge and innovative technologies for sustainable communities and environment.

Towards this end, we, at the Visayas State University, commit to:

- Produce highly competent, quality and world-class manpower in science and technology, especially for agriculture, environmental management and industry who are proficient in communication skills, critical thinking and analytical abilities;
- Generate and disseminate relevant knowledge and technologies that lead to improved productivity, profitability and sustainability in agriculture, environment and industry; and
- Satisfy the needs and applicable requirements of the industry, the community and government sectors who are in need of quality graduates and technology ready for commercialization through the establishment, operation, maintenance and continual improvement of a Quality Management System (QMS) which is aligned with the requirements of ISO 9001:2015.

It shall be the policy of the university that the quality policies and procedures are communicated to and understood by all faculty, staff, students and other stakeholders and that the system be continually improved for its relevance and effectiveness.

EDGARDO E. TULIN
President
v0 07-16-2019

4. Quality Goals of the College of Engineering and Technology
    a. Produce globally competent engineering graduates by providing students with excellent instruction through updated curriculum; functional and state-of-the art facilities; and qualified, well-trained, and dedicated faculty and staff;
    b. Generate new and advance knowledge and technology in engineering and allied sciences through the conduct of relevant researches that can contribute towards sustainable development, climate change mitigation, food security, and advance knowledge in engineering sciences; and
    c. Engage in relevant need-based community/stakeholder-projects that can make the Philippines and even the world a better place to live in.

5. Quality Objectives of the Department of Computer Science and Technology
    a. Graduates of the program are IT professionals and researchers, and proficient in designing and developing computing solutions.
    b. Excellent and relevant education in computer science and technology;
    c. Generate appropriate knowledge in Information and Communications Technology relevant to agricultural production, processing, utilization, technology generation and dissemination;
    d. Sustainable linkages and cooperation with public and private institutions in instruction, research and developments and extension.
    e. Dynamic linkages with other agencies and institutions for the promotion of instruction, research, extension programs in computer science and allied fields.
    f. Sustained linkages among the various units within the university to support developmental programs.

II. PROGRAM INFORMATION

1. Name of the Program	Bachelor of Science in Computer Science
2. CHED CMO Reference	CHED CMO No. 25 s. 2015
3. BOR Approval	BOR Resolution No. 76 s. 2018

4. Program Educational Objectives and Relationship to Institution Mission

Program Educational Objectives	Mission		
	a	b	c
1. Articulate and discuss the latest developments in the specific field of practice.	✓	✓	✓
2. Work effectively and independently in multi-disciplinary and multicultural teams.	✓	✓	✓
3. Analyze complex problems and identify and define the computing requirements needed to design an appropriate solution.	✓	✓	✓
4. Apply computing and other knowledge domains to address real-world problems.	✓	✓	✓
5. Design and develop computing solutions using a system-level perspective.	✓	✓	✓
6. Utilize modern computing tools.	✓	✓	✓
7. Those employed in industry or entrepreneurial endeavors will demonstrate professional advancement through expanded leadership responsibility, significant technical accomplishment, or other recognition of their contributions.	✓	✓	✓
8. Those who continue their formal education will achieve an advanced degree or other technical certification.	✓	✓	✓

a - development of a highly competitive human resource, b - cutting-edge scientific knowledge, c - innovative technologies for sustainable communities and environment

III. COURSE INFORMATION

1. Course Code	CSci 21
2. Course Title	Data Structures and Algorithms
3. Pre-requisite	CSci 14 – Intermediate Programming CSci 102 – Discrete Structures I
4. Co-requisite	None
5. Credit	3 units
6. Semester Offered	1st semester
7. Number of hours	2 hours lecture and 3 hours laboratory per week
8. Course Description	The course covers the standard data representation and algorithms to solve computing problems efficiently (*with respect to space requirements and time complexity*). This covers the following: arrays, linked lists, stacks, queues, and trees.

9. Program Outcomes (POs) **in relation to the Program Educational Objectives** (PE[Os)

Program Outcomes (POs)		**Program Educational Objectives**							
		1	2	3	4	5	6	7	8
a	Apply knowledge of computing fundamentals, knowledge of a computing specialization and mathematics, science and domain knowledge appropriate for this computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.			✓					
b	Identify, analyze, formulate, research literature and solve complex computing problems and requirements reaching substantiated conclusions using fundamental principles of mathematics, computing sciences and relevant domain disciplines.			✓	✓				
c	Apply mathematical foundations, algorithmic principles and computer science theory in the modelling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.			✓					
d	Knowledge and understanding of information security issues in relation to the design, development and use of information systems.				✓				
e	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.				✓	✓			
f	Create, select, adapt and apply appropriate techniques, resources and modern computing tools to complex computing activities with an understanding of the limitations to accomplish a common goal.						✓		
g	Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings.		✓						
h	Communicate effectively with the computing community and with society at large about complex								

	computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations and give and understand clear instructions.								
i	Recognize the legal, social, ethical and professional issues involved in the utilization of computer technology be guided by adoption of appropriate professional, ethical and legal practices.								
j	Recognize the need and the ability to engage in independent learning for continued development as a computing professional.	✓							✓

10. Course Outcomes (COs) and Relationship to Program Outcomes (POs)

After completing this course, the student must be able to perform the following COs:	*Program Outcomes Code*									
	a	b	c	d	e	f	g	h	i	j
CO1: Manage allocated dynamic memory using C	*E*	*E*	*E*		*I*	*I*				
CO2: Utilize arrays in solving computational problems	*E*	*E*	*E*		*I*	*I*				
CO3: Evaluate runtime complexity of algorithms	*E*	*E*	*E*							
CO4: Utilize linked lists in solving computational problems	*E*	*E*	*E*		*I*	*I*				
CO5: Use stack and queue operations to solve computational problems	*E*	*E*	*E*		*I*	*I*				
CO6: Use tree operations to solve computational problems	*E*	*E*	*E*		*I*	*I*				

Legend: I – Introductory, E – Enabling, D – Demonstrative
Each letter indicates the expected level of competency that each CO should provide for each PO.

11. Course Content and Plan

Week	Topics	Learning Outcomes	Teaching and Learning Activities		Assessment Tasks
			Teaching Activities	Learning Activities	
1	**Class Orientation** 1. OBE Syllabus 2. Requirements 3. Assessment System and Activities 4. Class Policies 5. Safety Guidelines and Emergency Response during fire and earthquake		• Setting of expectations • Agreements • Learning mode • Mode of communication • Familiarization of VSUEE/VC		
CO1: Manage allocated dynamic memory using C					
2-4	**Module 1: Memory Allocation and Management** Lesson 1.1: Memory Allocation	LO 1.1.1 Write valid C code that utilizes pointers LO 1.1.2 Visualize conceptually the configuration of user-defined variables at any point of the code implementation LO 1.1.3 Apply multiple levels of indirections using pointers correctly LO 1.1.4 Allocate memory dynamically	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	Module 1 Post-test, Laboratory Exam
	Lesson 1.2: Dynamic Memory Management	LO 1.2.1 Utilize void* in memory referencing LO 1.2.2 Use the sizeof operator to acquire the memory size allocated by a specific reference LO 1.2.3 Avoid memory leaks and dangling pointers when managing dynamic memory LO 1.2.4 Assert valid use of free function in deallocating dynamic memory			
CO2: Utilize arrays in solving computational problems					
4-6	**Module 2: Arrays** Lesson 2.1: Static Arrays	LO 2.1.1 Create and initialize static arrays LO 2.1.2 Access data in licit locations of the static array	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	Module 2 Post-test, Laboratory Exam

	Lesson 2.2: Dynamic and Jagged Arrays	LO 2.2.1 Create and initialize dynamic/jagged arrays LO 2.2.2 Access data in licit locations of the dynamic/jagged array			
	Lesson 2.3: Array as a Data Structure	LO 2.3.1 Utilize static and dynamic arrays as an efficient storage medium of data LO 2.3.2 Solve computing problems effectively using static/dynamic arrays			
CO3: Evaluate runtime complexity of algorithms					
7-8	**Module 3: Complexity Analysis** Lesson 3.1: Computational and Asymptotic Complexity	LO 3.1.1 Define Big O (O), Omega (Ω), Theta (Θ) notations LO 3.1.2 Differentiate computational from asymptotic complexity LO 3.1.3 Prove/disprove asymptotic complexity assumptions of a function/algorithm	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	Module 3 Post-test
	Lesson 3.2: Computing Asymptotic Complexity	LO 3.2.1 Estimate the asymptotic runtime complexity of a code snippet, function, or algorithm			
9	**MIDTERM EXAMINATION WEEK**				
CO4: Utilize linked lists in solving computational problems					
10-11	**Module 4: Linked Lists** Lesson 4.1: Singly Linked Lists	LO 4.1.1 Create and initialize singly linked lists LO 4.1.2 Perform insertion and deletion of elements in a singly linked list	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	Module 4 Post-test, Laboratory Exam
	Lesson 4.2: Doubly and Circular Linked Lists	LO 4.2.1 Create and initialize doubly and circular linked lists LO 4.2.2 Perform insertion and deletion of elements in a doubly and circular linked lists			
	Lesson 4.3: Linked Lists as a Data Structure	LO 4.3.1 Utilize linked list as an efficient storage medium of data LO 4.3.2 Solve computing problems effectively using linked lists			

CO5: Use stack and queue operations to solve computational problems					
12-13	**Module 5: Stack and Queue** Lesson 5.1: Stack	LO 5.1.1 Enumerate and define the basic operations of a stack ADT LO 5.1.2 Compute the asymptotic complexities of stack operations LO 5.1.3 Assert solutions on computing problems involving stack ADT	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	Module 5 Post-test, Laboratory Exam
	Lesson 5.2: Queue and Priority Queue	LO 5.2.1 Enumerate and define the basic operations of a queue ADT LO 5.2.2 Compute the asymptotic complexities of queue operations LO 5.2.3 Assert solutions on computing problems involving queue ADT			
CO6: Use tree operations to solve computational problems					
Module 6: Trees					
14-17	Lesson 6.1: Introduction to Binary Trees	LO 6.1.1 Convert conceptual binary tree representations into parenthetical notations and v.v. LO 6.1.2 Identify the valid properties of a binary tree LO 6.1.3 Perform preorder, inorder, and postorder traversals given a binary tree definition LO 6.1.4 Create a binary tree using predefined binary tree traversals LO 6.1.5 Compute the balance factor of a binary tree LO 6.1.6 Generate a binary expression tree given an infix expression LO 6.1.7 Synthesize a valid postfix expression from a binary expression tree	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	Module 6 Post-test, Laboratory Exam
	Lesson 6.2: Binary Search Trees	LO 6.2.1 Insert correctly an element into a binary search tree LO 6.2.2 Perform element search in a binary search tree LO 6.2.3 Delete an element	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	

		successfully in a binary search tree LO 6.2.4 State the advantage of BST over a generic binary tree			
	Lesson 6.3: AVL Trees	LO 6.3.1 Insert correctly an element into an AVL Tree LO 6.3.2 Delete an element successfully in an AVL Tree LO 6.3.3 Ensure that all instances of AVL Trees are always balanced	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	
	Lesson 6.4: B-Trees	LO 6.4.1 Insert correctly an element into a B-Tree LO 6.4.2 Delete an element successfully in an existing B-Tree LO 6.4.3 Compare the difference of the operations between binary search trees and B-trees in terms of its operations	• Coding demonstration • Interactive discussion of concepts	• Reading of electronic reading material • Group study among peers	
	STUDENT SELF-ASSESSMENT SURVEY				
18	**FINAL EXAMINATION WEEK**				

** VSUEE/VC – VSU E-Learning Environment/ Virtual Classroom*

12. Life-long Learning Opportunities

- Students' critical thinking and spatial abilities will be further developed since creating the architecture and implementation of the data structures will require to trace the step-by-step solution in order to achieve the task of a certain function or object
- The awareness of student's in considering the complexity of a specific algorithm will be raised

13. Contribution of Course to Meeting the Professional Component (%)

General Education: 0%
Basic ICT: 50%
Professional Computer Science*: 5*0%

14. References and Other Learning Resources

A. Textbook(s)

1. Barnett, Granville; del Tongo, Luca. *Data Structures and Algorithms: Annotated Reference with Examples*. First Edition. 2008.
2. Cormen, T.H.; Leiserson C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*. Third Edition. Massachusetts Institute of Technology, 2009.
3. Drozdek, Adam. *Data Structures and Algorithms in C++.* Fourth Edition. Cengage Learning, 2013.
4. Gilberg, Richard F.; Forouzan, Behrouz A. *Data Structures: A Pseudocode Approach with C.* Course Technology Thomson Learning, 2005.
5. Harris, S.; Ross, J. *Beginning Algorithms*. Wiley Publishing Inc., 2006.

6. Pfaff, B. *An Introduction to Binary Search Trees and Balanced Trees*. Free Software Foundation Inc., 2004.
7. Reese, R. *Understanding and Using C Pointers*. O'Reilly Media Inc., 2013.
8. Wengrow, J. *A Common-Sense Guide to Data Structures and Algorithms*. The Pragmatic Programmers LLC, 2017.

B. Online Resources

1. Data Structures, Hacker Earth
   https://www.hackerearth.com/practice/data-structures/
2. Data Structures, Geeks for Geeks
   https://www.geeksforgeeks.org/data-structure-gq/
3. Data Structures, Code Signal (Free Style Interview Practice)
   https://app.codesignal.com/interview-practice

15. Course Assessment and Evaluation

The performance of students will be assessed and evaluated based on the following:

50% Midterm + 50% Final Term = 100% (Overall Final)

Item No,	Assessment Tasks	Percentage Contribution (1)	No. of Times in the Semester (2)	Individual Task % Contribution (1/2)
1	Module Post-tests (MP)	30	6	5
2	Term Exams (TE)	40	2	20
3	Laboratory Exam (LE)	30	2	15
	TOTAL	**100%**		

COs	Assessment Tasks	Weight in Percent	Minimum Average for Satisfactory Rating	Target and Standards
CO 1	Module 1 Post-test	5	60 %	At least 50% of the students have at least 60% score
CO 2	Module 2 Post-test	5		
CO 3	Module 3 Post-test	5		
	Midterm Exam	20		
	Midterm Laboratory Exam	15		
CO 4	Module 4 Post-test	5		
CO 5	Module 5 Post-test	5		
CO 6	Module 6 Post-test	5		
	Final Exam	20		
	Final Laboratory Exam	15		
TOTAL		**100%**		

Grading System (% Passing: 60%)

Range	Grade	Range	Grade
95.56-100.00	1.00	52.50-59.99	3.25
91.11-95.55	1.25	45.00-52.49	3.50
86.67-91.10	1.50	37.50-44.99	3.75
82.22-86.66	1.75	30.00-37.49	4.00
77.78-82.21	2.00	22.50-29.99	4.25
73.33-77.77	2.25	15.00-22.49	4.50
68.89-73.32	2.50	07.50-14.99	4.75
64.44-68.88	2.75	00.00-07.49	5.00
60.00-64.43	3.00		

16. Course Policies

- **VSU E-Learning Portal** (https://elearning.vsu.edu.ph/) is the official learning portal for this course.
- If the pre-requisite course/s of this course is graded INC and not complied before Midterm Exam Week, the student will automatically be dropped from the course.
- This syllabus is designed for face-to-face lecture classes.
- Deadlines are absolute unless extended due to any force majeure e.g., natural disasters, hospitalization (inpatient), sickness (outpatient), war, suspension of classes via VSU OP memoranda. In the event of sickness (outpatient), it must be supported with a medical certificate signed by the doctor with his license number attached, otherwise the claim will not be accepted.
- University policies are strictly followed. Please be guided accordingly.
- Lastly, as we embark in this "new normal". Let us have an open mind and heart as we adjust in this new way of delivering the teaching-learning process and continue to aim for the highest quality in computer science education.

This class policy serves as our written agreement for the whole midyear. If there are any changes to enhance the class learning opportunity within the semester, it will be communicated accordingly.

17. Course Materials and Facilities Available

Faculty:	Student:
• Whiteboard, marker, and eraser • Laptop • C compiler (GCC) • Projector	• VSUEE (E-learning portal) • Electronic reading materials • PC • C compiler (GCC)

18. Revision History

Revision number	Date of Revision	Date of implementation	Highlights of Revision
00	December 28, 2018	August 1, 2019 1st semester, AY 2019-2020	New syllabus creation
01	October 2, 2019	N/A	ISO OBE syllabus format transfer
02	May 6, 2020	August 10, 2020 1st semester, AY 2020-2021	Revised for the blended mode of learning (pandemic season)
03	August 24, 2022	September 12, 2022 1st semester, AY 2022-2023	Revision of topics, assessment tasks, learning mode and grading system

19. Preparation

	Name	Signature	Date Signed
Prepared by	Jomari Joseph Barrera		

IV. INSTRUCTOR/PROFESSOR INFORMATION

1. Name of Instructor/Professor	Jomari Joseph A. Barrera
2. Office and Department	Department of Computer Science and Technology
3. Telephone/Mobile Numbers	(053) 565 0600 loc. 1022 / +63 970 068 4506
4. Email Address	jomarijoseph.barrera@vsu.edu.ph
5. Consultation Time	Monday and Wednesday 1 – 2 PM

20. Department Instructional Materials Review Committee:

Committee	Name	Signature	Date Signed
Member:	Eugene Val D. Mangaoang		
Member:	Michael Anthony Jay B. Regis		
Member:	Jude B. Rola		
Chairperson:	Jonah Flor O. Maaghop		

	Name	Signature	Date Signed
Verified by:	**JANNET C. BENCURE** College Dean		
Validated by:	**NANCY D. ABUNDA** Head, IMD		

Note:

1) The number of POs will depend on each degree program offered
2) COs and Relationship to POs
    a. (I) - **Introductory** – an Introductory Course to an outcome
    b. (E) - **Enabling** – an Enabling Course or a course that strengthens the outcome
    c. (D) - **Demonstrated** – a Demonstrative Course or a course demonstrating an outcome.

REMINDER:

1. *The author should not be part of the DIMRC.*
2. **If the author is the Department Head, he/she will be replaced by another chairperson from among the senior faculty members.*
3. ***If the author is the College Dean, the Head of Instructional Materials Development will approve.*
4. *Follow the next higher supervisor, no same person*
5. *For the component campuses, if the author is the College Dean, the Director for Academic Affairs will approve.*
6. *If the author is the Department Head and at the same time the College Dean, the Director for Academic Affairs will be the Chairperson of the DIMRC, and the Chancellor will approve it.*

(3) Distribution of copies: OHIMD, Department, Faculty