



**Bicol University  
Bicol University Polangui  
Polangui, Albay  
S.Y. 2024 - 2025**



---

# **DATA STRUCTURE AND ALGORITHM: PROJECT/FINAL PRESENTATION**

**Prepared by:**

**Dela Paz, Paula Bianca B.**

**Nobleza, Otelo P.**

**Sinfuego, Andrei Lloyd V.**

**Presented to:**

**Professor Khristine Botin**

**Bachelor of Science in Information System 2-B**

---



---

## Project Overview: Mini Digital Dictionary

The goal of this project is to develop a digital dictionary that covers the key lessons learned during the first semester finals. The primary objective is to provide a tool for reviewing and reinforcing concepts from the course, demonstrating our understanding of data structures such as binary trees, binary search trees, and heaps. By creating this program, we showcase our ability to apply these concepts in a practical, interactive way. When a user selects a topic, the program displays its definition and provides relevant explanations, ensuring clarity and helping the user better understand the subject. This project serves not only as a learning tool but also as a way to demonstrate our growth as developers. By writing the code ourselves, we apply what we've learned in class and create a useful tool for others to explore these data structures.

## How to Run the Program

1. **Prepare Your Environment:** Ensure you have a C++ compiler (e.g., GCC) or an IDE like Visual Studio, CLion, or Code::Blocks.
2. **Copy the Code:** Copy the full source code into a new C++ file and save it with a .cpp extension (e.g., mini\_digital\_dictionary.cpp).
3. **Compile the Program:** Compile the code using your chosen C++ compiler or IDE.
4. **Run the Program:** Once compiled, run the generated file. On Windows, you can double-click the executable or run it via Command Prompt.
5. **Interact with the Program:** The program will display a welcome message with a list of topics. Enter a topic to see its definition and details.
6. **Entering Array Values:** For topics like Heapify, Min Heap, and Max Heap, input values separated by spaces. Enter -1 to finish input.
7. **Arranged Output:** After entering values, the program will arrange them as specified (e.g., into a min-heap or max-heap) and display the result.
8. **Exit the Program:** You can continue exploring topics or type "exit" to quit the program.

## Program Functionality

This C++ program serves as an interactive mini digital dictionary that helps users explore key data structures, including Binary Trees, Binary Search Trees, Min Heaps, and Max Heaps. Users can input values to build these structures and perform operations such as traversals, searches, and deletions. For Binary Trees, the program supports inserting values and performing an inorder traversal to display the values in ascending order. The Binary Search Tree (BST) extends this by enforcing the property that each node's left child contains a smaller value and the right child contains a larger value, allowing for efficient search and deletion operations. When searching for a specific value in the BST, the program reports whether it is found or not. Additionally, the program allows users to heapify an array of values into either a Min Heap or Max Heap. A Min Heap ensures that the smallest element is at the root, while a Max Heap places the largest element at the root, with the program displaying the resulting heap structure after heapification. The program is designed to be interactive, where users can search for topics like "Binary Tree", "Binary Search Tree", "Heapify", "Min Heap", or "Max Heap", with the program providing a description of the data structure and guiding the user through operations related to the selected topic. It also supports case-insensitive input, ensuring that the program correctly processes search terms regardless of case. The loop allows users to perform multiple searches, and typing "exit" will terminate the program. Overall, the program functions as both an educational tool and a hands-on way for users to interact with and better understand these fundamental data structures and algorithms.

---