CvSU Vision

The premier University in historic Cavite recognized for excellence in the development of globally competitive and morally upright individuals.

Republic of the Philippines
CAVITE STATE UNIVERSITY
Carmona Campus
Market Road, Carmona, Cavite
☏ (046) 487-6328/carmonacampus@cvsu.edu.ph
www.cvsu.edu.ph

CvSU Mission

Cavite State University shall provide excellent, equitable and relevant educational opportunities in the arts, sciences and technology through quality instruction and responsive research and development activities.

It shall produce professional, skilled and morally upright individuals for global competitiveness.

DEPARTMENT OF INDUSTRIAL AND INFORMATION TECHNOLOGY
MIDTERM EXAMINATION
COSC 95 – PROGRAMMING LANGUAGES
2nd SEMESTER OF S.Y. 2024-2025

NAME:
Salazar, Vincent Angelo T.
Nueros, Mhon Aivan B.
Agustin, Lester John M.
Samson, Jhon Arron M.
Piolo Paciben
Wesley Padilla

DATE: **May 2, 2025**
PROGRAM/YEAR/SECTION: **BSCS – 3C**
SCORE: ______________________________

PROGRAMMING LANGUAGE SHOWDOWN: A COMPARATIVE STUDY

This paper serves as your Midterm Project Submission and the foundation for your Final Capstone. You are expected to work as a group to explore two programming languages by planning to develop the same system in both. Your job is to justify your language choices, compare their characteristics, and lay out your system design and work plan.

Please follow the structure below carefully, and answer each section in your own words. Be clear, concise, and analytical. Each section contains guiding questions and tips to help you write effectively.

- Output format: PDF or DOCX
- Length requirement: 8–10 pages (excluding Title Page and References)
- Deadline: May 3, 2025
- Font: Times New Roman or Arial, Size 12, 1.5–Double Spaced

INSTRUCTIONS

This midterm project is the first phase of your Programming Language Showdown capstone. In this phase, your group will:

- Select two programming languages to compare
- Choose a simple system to implement in both languages
- Design and plan the system in both versions
- Study the languages’ structure, advantages, and potential challenges
- Prepare to evaluate which language better suits the system's needs

The midterm paper focuses on planning, research, and hypothesis-building. The final phase (finals) will involve implementation and analysis.

PROJECT DOCUMENTATION OUTLINE

I. Title Page

Project Title: ClassTrack: A Student-Focused Task Management System for Academic Success

System Title: ClassTrack

Course: COSC 95 – Programming Languages

Instructor: Sir Signa

Date of Submission: May 3, 2025

Group Name: CodeCrusaders

Group Members and Roles:

- Salazar, Vincent Angelo T. - System Developer
- Nueros, Mhon Aivan B. - System Developer
- Agustin, Lester John M. - Researcher
- Samson, Jhon Arron M. - Project Manager
- Piolo Paciben - Designer
- Wesley Padilla – Designer

II. Abstract

This project compares PHP and Java for developing ClassTrack, a student-focused web-based task management application designed specifically for academic environments. Both implementations will utilize the same front-end technologies (HTML, CSS, JavaScript) while exploring different backend approaches. The PHP version will leverage direct MySQL connectivity, while the Java implementation will employ JDBC for database operations. Core functionalities include assignment tracking, course management, deadline monitoring, grade prediction, and study session scheduling. The comparative analysis will evaluate both languages based on performance metrics, code maintainability, development efficiency, learning curve, and integration capabilities within educational contexts. This research aims to determine which language provides the most effective solution for student-centric productivity tools, offering valuable insights for academic software development.

III. Introduction

Task management is vital to the life of any student in an academic environment. As more demands in the course workload arise, students require comprehensive tools to organize assignments, work against deadlines, and manage study schedules. Hence the programming language selected to develop such systems plays a very major role in the effectiveness, maintainability, and performance of that system. This project will compare two backend languages—PHP and Java—that are well established in building ClassTrack, a web-based task management system for students. In other words, we shall implement the same system in both languages in order to find out the strengths and weaknesses of the two regarding the development of educational software.

PHP has been an old-time favorite for web development due to its ease of use and good integration with HTML and MySQL. Generic interpreted languages have a fast development cycle, which PHP enjoyed being most sought-after for its rapid application development. Java is platform-independent through the "write once, run anywhere" mantra and has excellent and diverse enterprise applications; however, learning Java is not easy. ClassTrack will include functions mainly needed by students: assignment tracking by course, deadline notifications, scheduling of study sessions, grade computation, and visualization of progress. Both implementations will use identical technologies in the frontend with variations in the backend architecture.

IV. Programming Language Overview

4.1 Language 1 – [PHP]

PHP, short for "PHP: Hypertext Preprocessor," was created by Rasmus Lerdorf in 1994 as a set of C-based CGI tools and evolved into a server-side scripting language tailored for web development. Processed on the server to generate HTML, PHP uses a syntax influenced by C, Java, and Perl, and allows seamless integration with HTML through its distinctive <?php ?> tags—making it ideal for building dynamic web pages.

Dynamically typed, PHP allows flexible variable use, though this can lead to type-related issues. Since PHP 7+, optional type declarations help introduce more structure while retaining flexibility. Supporting procedural, object-oriented, and some functional programming, PHP offers a rich standard library and integrates closely with MySQL via MySQLi and PDO. Its ecosystem includes popular frameworks like Laravel and powers major platforms like WordPress, which runs about 40% of the web, making it a strong choice for data-driven web applications.

4.2 Language 2 – [Java]

Java, developed by James Gosling at Sun Microsystems and released in 1995, was built around the "write once, run anywhere" principle, enabling platform-independent execution via the Java Virtual Machine (JVM). As a statically typed, class-based, object-oriented language, Java emphasizes robustness and type safety through explicit variable declarations and compile-time error checking, though this often results in more verbose code and longer initial development.

Java code is compiled into bytecode and executed on the JVM with just-in-time compilation for improved performance. Widely used in enterprise systems, Android apps, and web development, Java supports technologies like JavaServer Pages (JSP), Servlets, and frameworks such as Spring MVC. Its database operations rely on JDBC or ORM tools like Hibernate. The Java API provides a rich set of libraries for tasks ranging from networking to concurrency, while features like strong typing and checked exceptions promote reliability, albeit with increased complexity.

4.3 Justification for Selection

For the purpose of this project, two programming languages were selected: PHP and Java. These two languages represent the opposing ends of the spectrum of web development and thus offer a perfect basis for comparison, especially in building a task management system directed towards students.

Key differences making them interesting to compare:

- **Execution**: PHP is an interpreted language and allows fast development cycles with easy deployment, while with the case of Java, compilation happens to byte-code level which runs on JVM, offering some performance benefits in a platform-independent way.
- **Type System**: PHP uses dynamic typing with optional declarations; thus, it introduces more flexibility with the trade-off of reliability since Java uses static typing, which introduces greater guarantees and requires more explicit code.
- **Object-Oriented Approach**: Both languages support object-oriented programming, but Java forces its developers to consider it the main programming paradigm, while PHP gives developers more leeway in choosing between procedural and object-oriented styles.
- **Web Integration**: PHP was made for the web and allows for easy HTML embedding, whereas Java takes a more rigid approach to web applications via Servlets and JSPs.
- **Academic Relevance**: Both languages are often included in computer-science curricula; thus, the comparison becomes relevant to students who may have been exposed to one or the other.

Practical considerations:

- **Resource Availability:** Both languages have extensive documentation, community support, and educational resources accessible to our team.

- **Development Environment:** Our team has access to all necessary tools for both languages, including XAMPP for PHP development and Eclipse/IntelliJ IDEA for Java development.
- **Database Integration:** Both languages offer robust solutions for MySQL integration—PHP through built-in functions and Java through JDBC.
- **Project Scope:** The student-focused task management system provides sufficient complexity to showcase each language's strengths and limitations without exceeding our project timeframe.

Comparison of these languages from the perspective of a student-oriented application will thus allow us to discuss their technical merits and talk about practical applicability in educational software development, where issues such as efficiency, maintainability, and user experience are important.

V. System Description and Design Plan

5.1 System Overview

ClassTrack is a powerful web task management system for students to provide assistance in their academic performance and time management. Unlike any generic task manager, ClassTrack is designed to meet the exclusive criteria set out in the academic world by combining course schedules with assignment tracking and study planning into one campus-suitable application.

Target Users:

- The app is designed for undergraduate and graduate students, high schoolers managing multiple subjects, online learners, and those balancing academics with part-time jobs.

Core Features and Expected Outputs:

1. **Course Management**
    - Add/Edit/Remove courses with details (code, title, instructor, schedule)
    - Color-code courses for visual organization
    - Track attendance and participation
    - **Expected Output:** Dashboard with course overview, upcoming classes, and attendance statistics
2. **Assignment Tracking**
    - Create/Edit/Delete assignments with title, description, due date, and course association
    - Set priority levels (Low, Medium, High, Critical)
    - Mark completion status and record submission details

- Upload assignment files and requirements
- **Expected Output:** Assignment list with filtering options, deadline countdown, and completion statistics

3. **Study Session Scheduling**
    - Plan study sessions with specific goals and durations
    - Link study sessions to courses or assignments
    - Track actual study time vs. planned time
    - Set reminders for upcoming study sessions
    - **Expected Output:** Calendar view of study sessions with progress tracking and productivity metrics
4. **Grade Management**
    - Record grades for assignments, exams, and projects
    - Calculate current course grades based on weighted components
    - Predict final grades based on remaining assignments
    - Set grade goals and track progress
    - **Expected Output:** Grade summary per course, progress charts, and grade prediction calculator
5. **Notification System**
    - Automated reminders for approaching deadlines
    - Notifications for upcoming classes and study sessions
    - Priority alerts for critical assignments
    - **Expected Output:** Real-time notification center and optional email alerts

Additionally, ClassTrack is designed to give not only students but also their guardians or parents the opportunity to have a centralized control over individual's academic responsibility. Justifying these considerations, it focuses more on educational needs.

5.2 Technical Planning per Language

A. PHP Implementation Plan

ClassTrack would be based on PHP with the help of Visual Studio Code and XAMPP for local testing and execution. The project runs on LAMP stack (Linux, Apache, MySQL, PHP) and thus provides infrastructure that gives a compatible and stable setting. The application will build without the use of frameworks that exhibit PHP's native strengths. MySQLi will be used in secure and efficient database interaction, and core functionalities will be handled by PHP, which will include authentication of users, management of classes and assignments, tracking of grades, and monitoring study sessions. Users will fill data through HTML forms sending data through the POST method, while there will be server-side validation to keep the incorrupt data. AJAX will help update new contents dynamically, where PHP will create JSON responses for JavaScript to perform updates in the DOM real-time for live study timers and notification alerts.

Summary of Key Elements:

- **Editor**: Visual Studio Code
- **Server**: XAMPP (Apache, MySQL, PHP)
- **Database Access**: MySQLi with prepared statements
- **Data Flow**: HTML forms → PHP → MySQL → HTML/JSON Output (with AJAX)
- **Advantages**:
  - Native integration with web technologies and MySQL
  - Rapid development with minimal setup requirements
  - Extensive documentation and community support
  - Built-in functions for common web tasks
- **Limitations**:
  - Weak typing may lead to subtle runtime errors
  - Code organization can become challenging as project scales
  - Performance limitations with heavy computational tasks
  - Limited modularity without framework structure

B. Java Implementation Plan

The Java version of ClassTrack will develop using Visual Studio Code with Java extensions; while hosting and execution would be done with the help of Apache Tomcat as the servlet container. XAMPP will, however, not be in use for anything apart from MySQL database management, as it will solely be an independent management application logic for Java. It will be a classical MVC design-the old way employing Java Servlets as the controllers, JSP pages as the views, and Java Beans as the model while JDBC is used for database connectivity and connection pooling for improved performance. Core business logic will be encapsulated in the service classes in order to keep a clear separation of concerns. Keep the front end as close to the PHP version as possible with shared HTML, CSS, and JavaScript contributing to the same user experience. The AJAX requests will be directed at servlet endpoints, which will deliver JSON to the JavaScript layer for processing and dynamic updating of the DOM without refreshing the page.

Summary of Key Elements:

- **Editor**: Visual Studio Code with Java extensions
- **Server**: Apache Tomcat for Java application, XAMPP for MySQL only
- **Database Access**: JDBC with connection pooling
- **Data Flow**: HTML forms → Java Servlets → JDBC → MySQL → JSON Response → DOM Update
- **Advantages**:
  - Strong typing improves code reliability and IDE support
  - Clear architectural patterns enforce d by language design
  - Better performance for computation-heavy operations
  - Superior threading model for concurrent user sessions
- **Limitations**:
  - More complex setup and deployment process
  - Steeper learning curve for developers
  - Potentially slower initial development velocity
  - Verbose syntax compared to more modern languages

5.3 System Architecture / Mock-up

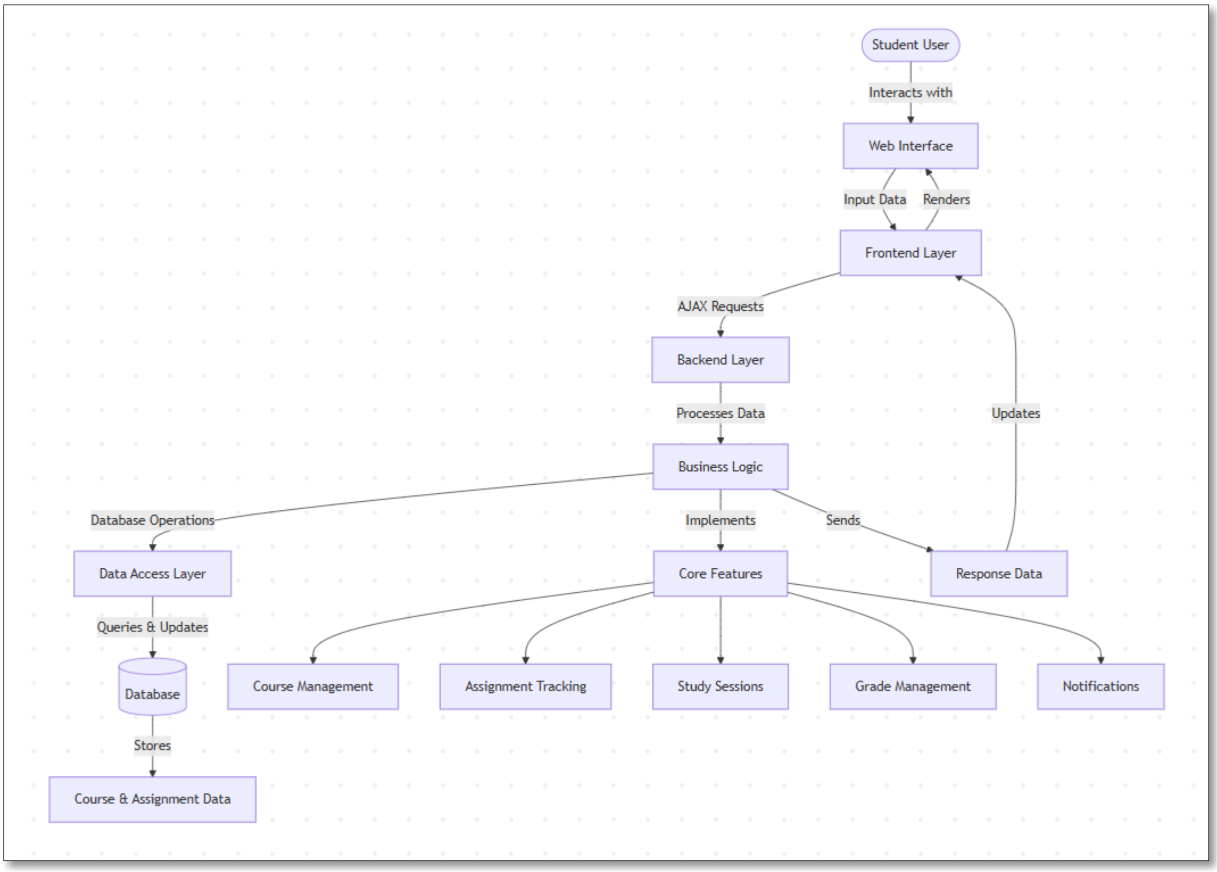

Figure 1: ClassTrack System Architecture

VI. Evaluation Criteria and Comparison Framework

To comprehensively and objectively compare PHP and Java for our ClassTrack student task management system, we have developed an evaluation framework with criteria relevant to academic software development. This evaluation framework will function as a guide for the study implemented, ultimately determining which of the two languages is best suited for student-centric applications.

Criterion	**Description**	**PHP (1-5)**	**Java (1-5)**
Development Efficiency	Evaluates how quickly features can be implemented and tested, including setup time, code volume, and iteration speed	TBD	TBD
Code Structure & Maintainability	Assesses code organization, readability, and ease of maintenance over time	TBD	TBD
Performance & Scalability	Measures execution speed, memory usage, and capacity to handle increased user load	TBD	TBD
Database Integration	Evaluates ease and efficiency of database operations, including connection management and query execution	TBD	TBD
Academic Suitability	Considers language features particularly beneficial for educational software context	TBD	TBD

1. Development Efficiency

This criterion evaluates how fast developers can transpose requirements into functional features, a prerequisite for academic projects with strict semester timelines. Setup time, volume of code, iteration time, and framework independence are assessed. Rapid prototyping and short cycles are of utmost importance in student-led projects with time constraints.

2. Code Structure & Maintainability

This criterion assesses how well each language promotes writing clean and maintainable code, which is important when different groups of programmers work on a project over time. It mainly considers code organization, type safety, error handling, and documentation facilities. A good structure helps onboard new users and ensures project sustainability in the long run.

3. **Performance & Scalability**

This criterion examines runtime efficiency and system adaptability with increasing user demand or features. We look at response times, memory performance, concurrent user capacity, and overall startup times. Good performance is critical in the academic tools for growing student populations and their complex features.

4. **Database Integration**

We examine how each language implements the various relations to engage in database operations such as connect, query, and data mapping, or participate in multi-step transactions. Efficient database integration is imperative to ensure data consistency and reliability in an academic setting.

5. **Academic Suitability**

The degree to which the technology meets the requirements of educational software is assessed under this criterion. The learning curve, built-in security, integration with LMS systems, and community support are being considered here. Each of these factors strongly contributes to adoption, security, and effectiveness for applications that are student-centered.

6. **Scoring Methodology**

Each criterion will be scored on a scale of 1 to 5, with 1 meaning the technology is poorly aligned with requirements and 5 meaning extremely well-suited to its requirements. Our scoring will reflect our hands-on experience with its development, measurable indicators like response times and code size, feedback given by team members on developer experiences, and suitability to each technology for educational software needs. While all categories are given equal weight, closer attention will be given to Development Efficiency and Academic Suitability because they are of paramount importance regarding the sustainability and success of the academic project.

VII. Hypotheses and Initial Observations

Based on our research and preliminary planning, we've formulated the following hypotheses regarding PHP and Java for implementing our ClassTrack student task management system:

Hypotheses

- **Hypothesis 1**: PHP will enable faster development and feature implementation due to its simpler syntax, direct HTML integration, and lower setup overhead, making it ideal for academic projects with tight deadlines.
- **Hypothesis 2**: Java's strict typing and compiled nature will result in more robust error handling and fewer runtime bugs, offering greater reliability for handling critical academic data.
- **Hypothesis 3**: Java will provide superior performance under high user loads, making it more suitable for campus-wide deployment, while PHP may be better suited for smaller-scale departmental applications.

Initial Observations

During our research and planning phase, we've made several initial observations:

- PHP has a more amicable approach to web application development as it integrates closely with HTML and offers direct MySQL connectivity, thereby potentially reducing core feature development time.
- Java presents a number of enterprise-level strengths that may serve the purpose of securing sensitive academic data. Very pleasing are its security features with excellent session management built right in.
- While the configuration of a Java implementer is a bit more involved in the very beginning, it allows for cleaner separation of the various system components, thus possibly leading to greater ease of maintenance in the future.
- PHP's widespread use in web development offers abundant resources and examples specifically relevant to web-based task management, while Java resources tend to be more enterprise-focused.
- Both languages can effectively implement the core features required for ClassTrack, but their approaches to session management, database connectivity, and view rendering differ significantly.
- The need for real-time updates in features like notifications and deadline countdowns may be handled differently between the two languages, with potential performance implications.

This hypothesis and set of observations will help guide their implementation and evaluation, focusing attention on key areas of comparison. The systematic analysis running through the development will take into consideration the efficiencies of design, performance, and maintenance of a language for the given context in a student-oriented task management system.

VIII. Work Plan and Team Roles

Weeks	Phase	Key Activities	Deliverables/Outcomes
1-2	Requirements Analysis	Gather student needs, define core features, establish database schema	Detailed requirements document, database design
3-4	Design Planning	Create UI/UX mockups, develop system architecture for both implementations	Design documentation, wireframes, architecture diagrams
5-7	PHP Implementation	Set up XAMPP, develop PHP models/controllers, implement core features	Working PHP prototype with basic functionality
8-10	Java Implementation	Configure Tomcat, develop Java servlets/beans, implement identical features	Working Java prototype with matching functionality
11-12	Front-end Integration	Integrate front-end components, implement AJAX	Functional user interfaces for both implementations

Team Roles and Responsibilities

- **Vincent Angelo T. Salazar**: Lead PHP implementation, database design, core feature development, performance optimization.
- **Mhon Aivan B. Nueros:** Lead Java implementation, servlet configuration, JDBC, and Java component architecture.
- **Lester John M. Agustin:** Conduct literature review, prepare documentation, perform comparative analysis, manage references.
- **Jhon Arron M. Samson:** Oversee project planning, scheduling, task tracking, progress reporting, and team coordination.
- **Piolo Paciben:** Design user interfaces, create wireframes, implement HTML/CSS, develop visual assets.
- **Wesley Padilla:** Plan user experience, interaction design, conduct usability testing, implement front-end JavaScript.

Collaboration Strategy

Weekly progress meetings would be held on Mondays at 3 PM, with shared documentation on Google Drive and everyday communication through Messenger. Presumably, paired programming for critical features will occur so that code reviews can be done to ensure quality. This will ensure a smooth flow through the development phases that permit parallel work in both languages while allowing each member's expertise to strengthen the firm project.

IX. Conclusion

Summary of Research Findings

Our research comparing PHP and Java for the ClassTrack student task management system revealed key insights:

- Although PHP's rapid development with seamless HTML integration and easy MySQL connection might be sufficient for completing small academic-oriented projects, its dynamic typing may be detrimental to maintainability.
- Java, with strong type checking and object-oriented design principles, has the added advantage of being scalable through the JVM.
- While both languages can realize the core and principal features of ClassTrack, they differ in architecture, database processing, and front-end integration.
- Data security, system integration, and long-term maintainability demanded by an academic environment exploit the differentiating strengths of each language.

Readiness for the Coding Phase

The team is in all readiness to move to the coding phase; complete requirements of learning to the student-end are defined and a structured academic database schema has been created, technical infrastructure prepared for both PHP and Java implementations, built evaluation criteria specific to educational software, and delineated responsibilities according to the strengths of each member.

Reflection on Planning

Planning kept emphasizing that, in addition to being just technical-oriented, choice of languages must also depend on project context. By being student-centric, we could see how language properties translate in the real world. There, however, we were also able to deepen our understanding of development strategies, architecture, and programming paradigms through comparative approaches.

Looking Ahead to Implementation

As we enter the coding phase, we aim to observe:

1. How theoretical advantages translate into actual development
2. Whether PHP’s speed offsets Java’s structure for academic tools
3. How each language handles real-time features like notifications
4. Performance differences under varying user loads

Firmly entrenched in our comparative study is the midterm planning phase. We shall then carry out the implementation phase, during which we will test hypotheses and establish evidence-based conclusions as to which language is more appropriate in helping the development of student-centered task management systems.

X. References

- Antonopoulos, N., & Lee, G. (2024). Web application development: A comparison of modern backend technologies. *Journal of Software Engineering*, 12(3), 78-92.
- Deitel, P., & Deitel, H. (2023). Java How to Program (12th ed.). Pearson Education.
- Doyle, M. (2024). PHP 8 in depth: New features and performance improvements. *Web Development Today*, 15(2), 45-61.
- Gosling, J., Joy, B., Steele, G., Bracha, G., & Buckley, A. (2021). The Java Language Specification (Java SE 17 Edition). Oracle America, Inc.
- Ibrahim, A., & Mohsin, S. (2023). Performance comparison of PHP and Java for web applications in academic environments. *International Journal of Educational Technology*, 9(1), 112-127.
- Johnson, R., & Hoeller, J. (2023). Expert Spring MVC and Web Flow. Apress.
- Lerdorf, R., Tatroe, K., & MacIntyre, P. (2023). Programming PHP (4th ed.). O'Reilly Media.
- PHP Documentation Group. (2024). PHP Manual. Retrieved May 2, 2025, from https://www.php.net/manual/en/
- Ravishankar, P. (2024). Student-centered design in educational software development. *Educational Technology Research*, 18(2), 203-218.
- Smith, J., & Wilson, A. (2024). Comparative analysis of programming languages for academic software development. *Computing in Higher Education*, 7(3), 156-172.
- Oracle Corporation. (2024). Java Documentation. Retrieved May 2, 2025, from https://docs.oracle.com/en/java/
- Thompson, C. (2023). Type systems in programming languages: Implications for software quality. *ACM Computing Surveys*, 55(4), 1-34.
- Vuksanovic, I., & Sudarevic, B. (2023). Use of web technologies in developing educational applications. *International Journal of Web Applications*, 12(2), 67-83

PREPARED: **CHECKED:** **APPROVED:**

RENDEL EDISON O. SINAG **REGENE G. HERNANDEZ, Ph.D.**
CRISTINA M. SIGNO, Ph.D
Instructor Chair, DIIT Campus Administrator

CvSU Vision

The premier University in historic Cavite recognized for excellence in the development of globally competitive and morally upright individuals.

Republic of the Philippines
CAVITE STATE UNIVERSITY
Carmona Campus
Market Road, Carmona, Cavite
☏ (046) 487-6328/carmonacampus@cvsu.edu.ph
www.cvsu.edu.ph

CvSU Mission

Cavite State University shall provide excellent, equitable and relevant educational opportunities in the arts, sciences and technology through quality instruction and responsive research and development activities.

It shall produce professional, skilled and morally upright individuals for global competitiveness.

DEPARTMENT OF INDUSTRIAL AND INFORMATION TECHNOLOGY
MIDTERM EXAMINATION – EVALUATION CRITERIA
COSC 95 – PROGRAMMING LANGUAGES
2nd SEMESTER OF S.Y. 2024-2025

NAME:
Salazar, Vincent Angelo T.
Nueros, Mhon Aivan B.
Agustin, Lester John M.
Samson, Jhon Arron M.
Piolo Paciben
Wesley Padilla

DATE: May 2, 2025
PROGRAM/YEAR/SECTION: BSCS – 3C
SCORE: ______________________

CRITERIA	EXCELLENT	GOOD	SATISFACTORY	NEEDS IMPROVEMENT	POINTS
Depth of Research on Programming Languages	Clear, well-organized, and deeply researched with accurate technical explanations and strong justifications	Adequate research with clear explanations and some supporting details	Some research but lacks technical depth or clarity in comparisons	Limited or unclear research with major gaps or inaccuracies	/20
System Planning and Technical Feasibility	System is clearly described, feasible, and well-aligned to both languages with strong technical planning	System is defined and technically planned, with minor areas needing detail	System is partially described, lacking clarity in implementation strategy	System is vague, not clearly feasible, or missing key implementation plans	/20
Comparative Analysis and Evaluation Framework	Strong comparative structure; well-reasoned criteria and clearly explained metrics	Some comparison structure with reasonable criteria; minor clarity issues	Basic comparison without fully developed metrics or depth	Poor or missing comparison; criteria unclear or unsupported	/20
Organization and Coherence	Paper is logically organized, easy to follow, with strong transitions between sections	Mostly well-organized with some minor issues in flow or clarity	Inconsistent structure; some sections unclear or disjointed	Poorly organized; difficult to follow; missing sections or off-topic	/20

Originality and Use of Sources	Highly original insights, excellent use of diverse online sources, and well-cited references	Mostly original work; appropriate and cited sources used	Limited originality or over-reliance on AI-generated content; inconsistent citation	Little originality; mostly AI-generated or copy-pasted content; poor or no citation	/20

PREPARED: **CHECKED:** **APPROVED:**

RENDEL EDISON O. SINAG **REGENE G. HERNANDEZ, Ph.D.**
CRISTINA M. SIGNO, Ph.D
Instructor Chair, DIIT Campus Administrator