Analyzing the Contribution of Programming and Database Courses to Capstone Performance Using Multiple Linear Regression

Findra Kartika Sari Dewi[1]

[1]Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta

E-mail: findra.dewi@uajy.ac.id[1]

Abstract. Capstone courses in software engineering require students to integrate knowledge and skills acquired from multiple prerequisite courses. However, not all prerequisite courses contribute equally to student success in capstone projects. This study investigates the contribution of individual programming and database courses to student performance in a software development capstone course using Multiple Linear Regression (MLR). Academic records from 482 undergraduate students were analyzed, including grades from prerequisite programming and database courses, with Grade Point Average (GPA) included as a control variable. Ordinary Least Squares (OLS) was used to estimate the regression model, while LASSO regression and partial correlation analysis were applied as supporting analyses to assess robustness and interpret direct relationships. The results indicate that Data Structures, Object-Oriented Programming, Web Programming, Software Project Management, and Platform-Based Programming have a significant positive contribution to capstone performance. In contrast, introductory courses show mainly indirect effects. Database Systems exhibits a reduced unique contribution after controlling for other courses, suggesting that its impact is embedded within broader development skills. These findings demonstrate that course-level academic data can provide actionable insights for curriculum evaluation and support data-driven improvements in capstone preparation.

Keywords: Capstone course; Multiple linear regression; Educational data mining; Programming courses; Academic performance.

1. Introduction

Capstone courses play a central role in computer science and software engineering education. These courses are designed to integrate knowledge and skills acquired from previous coursework into a comprehensive, project-based learning experience that resembles real-world software development practice. Through capstone projects, students are expected to demonstrate technical competence, problem-solving ability, and the capacity to apply prior learning in complex and open-ended tasks [1], [2].

Despite their importance, many institutions report substantial variation in student performance in capstone courses. Even students with similar academic backgrounds may experience different levels of success in capstone projects, reflecting substantial variation in capstone structure, pedagogy, and assessment across institutions [3]. This variation raises an important academic question: to what extent does performance in prerequisite courses contribute to student success in capstone projects?

Previous studies have shown that prior academic achievement is related to performance in advanced computing courses. In particular, programming-related courses are often identified as strong predictors of later success, as they provide essential skills for algorithmic thinking, software design, and implementation [4], [5]. Similarly, database-related knowledge plays an important role in software development education, particularly in modeling, data management, and system integration tasks that support applied software projects [6].

Several researchers have applied Educational Data Mining (EDM) techniques to predict student performance using academic records. Regression-based approaches, including Multiple Linear Regression (MLR), have been widely used due to their interpretability and suitability for analyzing academic factors [7], [8], [9]. In the context of capstone courses, [10] demonstrated that grades from prerequisite computing courses can be used to predict student performance in a software engineering capstone course with reasonable accuracy.

Most existing studies rely on aggregate indicators such as GPA or emphasize non-academic factors, while the specific contribution of individual programming and database courses to capstone performance remains underexplored. This gap is particularly relevant in software engineering curricula that include multiple programming courses with different emphases. Therefore, this study examines the relationship between student performance in prerequisite programming and database courses and achievement in a capstone software development course using Multiple Linear Regression, with the aim of identifying courses that have a significant and unique contribution to capstone success.

2. Literature Review

2.1. Capstone Courses in Software Engineering Education

Capstone courses are a key component of software engineering and computer science curricula, designed to integrate knowledge and skills from prior coursework into project-based learning that reflects real-world software development practice [1], [2]. Offered in the final year, capstone projects require students to apply competencies from multiple prerequisite courses, including programming, software design, and database systems.

Several studies have examined capstone course outcomes. Survey-based studies have shown that capstone courses vary widely in structure and implementation, with differences in project scope, assessment methods, and instructional support influencing student learning experiences[3]. Licorish et al. [11] also found that student effort, technical skills, and prior learning experiences are closely related to the quality of software artifacts produced in capstone projects.

However, most existing research focuses on overall learning outcomes or teamwork, with limited attention to the contribution of individual prerequisite courses. As a result, the specific role of programming and database courses in shaping capstone performance remains underexplored.

2.2. Programming and Database Courses as Prerequisites

Programming courses form the foundation of software engineering education by developing core skills in problem solving, abstraction, and software implementation. Empirical studies show that student performance in programming-related courses is strongly associated with success in advanced computing subjects [4], [5]. In particular, insufficient mastery of programming fundamentals has been linked to increased risk of failure in later courses.

Database courses also play an important role in software development curricula, especially in system design and data management. Previous studies have reported challenges in ensuring that database concepts, such as data modeling and querying, are consistently applied in broader software development contexts, even when database courses include practical activities[6].

Despite their recognized importance, prior studies rarely isolate the contribution of individual programming and database courses. Instead, programming ability is commonly treated as a general construct, limiting insight into which specific courses most strongly influence capstone performance.

2.3. Educational Data Mining for Academic Performance Prediction

Educational Data Mining (EDM) has been widely applied to analyze student academic performance and identify factors related to learning outcomes. Regression-based approaches, particularly Multiple Linear Regression (MLR), are commonly used due to their interpretability and suitability for structured academic data[7], [9].

Several studies have demonstrated the effectiveness of MLR for predicting student performance using course grades and academic records. Yang et al. [8] showed that regression-based models remain reliable when combined with feature reduction techniques. Prior reviews of educational data mining research have shown that regression-based approaches are widely used for predicting student academic performance, particularly when interpretability and understanding relationships among academic variables are prioritized[12].

Recent studies also indicate that regression-based models can achieve competitive performance compared to more complex approaches when academic variables are clearly defined [13], [14]. These findings support the use of MLR in studies that prioritize interpretability and course-level analysis, as in the present research.

2.4. Performance Prediction in Capstone Courses

In the specific context of capstone courses, [10] applied regression-based educational data mining to predict student performance in a software engineering capstone course. Their results showed that grades from prerequisite computing courses, particularly intermediate programming courses, are significant predictors of capstone success. This study provides empirical evidence that course-level academic data can be used to model capstone performance.

Other studies have explored additional factors influencing capstone outcomes, such as teamwork, project management, and student motivation [11], [15]. While these factors are important, they are often difficult to quantify consistently across cohorts and institutions. As a result, their use in predictive models is limited for curriculum-level analysis.

Taken together, prior research suggests that prerequisite programming and database courses play an important role in capstone performance, and that regression-based models are suitable tools for analyzing these relationships. However, there remains a lack of studies that systematically examine the individual contribution of multiple programming and database courses within a single regression framework.

2.5. Research Gap

Based on the reviewed literature, two main research gaps can be identified. First, although programming and database skills are widely recognized as essential for capstone success, their contributions are rarely analyzed at the level of individual courses. Second, while regression-based EDM methods are commonly used, they are often applied to aggregate indicators such as GPA rather than detailed course-level data.

This study addresses these gaps by applying Multiple Linear Regression to analyze the unique contribution of individual programming and database courses to student performance in a capstone software development course. By focusing on interpretable statistical relationships, the study aims to provide evidence that can directly inform curriculum evaluation and improvement.

3. Research Methodology

3.1. Research Design

This study employs a quantitative Educational Data Mining approach using Multiple Linear Regression (MLR) as the primary analytical model. MLR is selected because it enables simultaneous analysis of multiple academic variables while providing interpretable coefficients suitable for curriculum-level evaluation. Supporting analyses are applied to assess the robustness and consistency of the regression results.

3.2. Data Source

The dataset consists of academic records from undergraduate students enrolled in an Informatics program who have completed the capstone course *Software Development Project* (SDP). After data cleaning and preprocessing, records from 482 students were included in the analysis.

All data were obtained from the institution's official academic information system and anonymized prior to analysis. The dataset includes final grades from prerequisite programming and database courses, as well as the final grade of the capstone course.

3.3. Variables

The dependent variable in this study is Capstone Performance, defined as the final grade obtained by students in the *Software Development Project* (SDP) course, which represents overall performance in the capstone project. The independent variables consist of grades from prerequisite courses that are commonly associated with software development skills, including Basic Programming (BP), Data Structures (DS), Database Systems (DB), Software Engineering (SWE), Object-Oriented Programming (OOP), Software Project Management (SPM), Platform-Based Programming (PBP), and Web Programming (WP). These courses were selected because they represent core competencies required for software development and successful completion of capstone projects, as suggested by prior studies [4], [6], [11].

In addition, Grade Point Average (GPA) is included as a control variable to account for general academic ability. GPA is not treated as the main focus of analysis but is used to isolate the specific contribution of individual courses to capstone performance.

3.4. Data Preprocessing

Several preprocessing steps were conducted to ensure data quality and suitability for regression analysis:

1. Missing Data Handling: Records with incomplete or invalid grades were removed.
2. Outlier Inspection: Grade distributions were examined to identify extreme values. No severe outliers requiring transformation were detected.
3. Correlation Analysis: Pairwise correlations among independent variables were examined to identify initial relationships among courses and to detect possible overlap between predictors.
4. Multicollinearity Check: Variance Inflation Factor (VIF) was calculated to ensure that the regression coefficients were not strongly distorted by high correlations among predictors.

3.5. Multiple Linear Regression Model

The primary analytical model used in this study is Multiple Linear Regression, estimated using the Ordinary Least Squares (OLS) method. OLS is the standard estimation technique for MLR and is widely used in educational data mining due to its simplicity and interpretability [7], [8].

The general form of the regression model is defined as in the equation (1).

$$Y = \beta_0 + \beta_1 DP + \beta_2 ISD + \beta_3 BD + \beta_4 RPL + \beta_5 PBO + \beta_6 MPPL + \beta_7 PBP + \beta_8 PW + \beta_9 GPA + \varepsilon \qquad (1)$$

where **Y** represents capstone performance, measured by the final grade of the Software Development Project (SDP) course; $\boldsymbol{\beta_0}$ is the intercept; $\boldsymbol{\beta_i}$ represents the regression coefficient of each predictor; $\mathbf{X_i}$ represents each prerequisite course grade or GPA; and $\boldsymbol{\varepsilon}$ is the error term.

3.6. Supporting Analyses

To support the main regression analysis, LASSO regression and partial correlation analysis are applied as complementary methods. LASSO is used to check whether the main predictors remain stable when weak predictors are reduced. Partial correlation is used to examine whether each course still has a direct relationship with capstone performance after other courses are controlled.

Regularization-based regression techniques are commonly used in educational data mining to assess predictor stability and reduce redundancy among academic variables[12]. This analysis is not intended to replace Multiple Linear Regression but to validate the robustness of the main findings.

Partial correlation analysis is employed to examine the direct relationship between each prerequisite course and capstone performance while controlling for other variables. This analysis helps clarify relationships that may be affected by overlapping effects among courses, particularly in identifying changes between simple and controlled associations.

3.7. Model Evaluation

Model evaluation was conducted to assess whether the regression model can predict capstone performance on unseen data, not only explain the training data. Model performance is evaluated using the following metrics:

- Coefficient of Determination (R^2): Measures the proportion of variance in capstone performance explained by the model.
- Root Mean Squared Error (RMSE): Indicates the average magnitude of prediction error.
- Mean Absolute Error (MAE): Represents the average absolute difference between predicted and actual values.

To assess predictive capability, a holdout validation strategy is applied using an 80:20 train–test split. A simple baseline model based on the mean capstone score is included for comparison.

3.8. Summary of Methodology

This methodology is designed to ensure that the analysis remains focused on the research objective: identifying the contribution of individual programming and database courses to capstone performance. Multiple Linear Regression serves as the core analytical approach, while LASSO regression and partial correlation analysis provide additional validation and interpretive support.

4. Results and Analysis

4.1. Descriptive Statistics

This section presents an overview of the dataset used in this study. After preprocessing, a total of 482 student records were retained for analysis. The dataset includes grades from prerequisite programming and database courses, GPA as a control variable, and the final grade of the capstone course (SDP).

Table 1 summarizes the descriptive statistics of all variables, including the mean, standard deviation, minimum, and maximum values. Overall, students achieved relatively high average scores across most prerequisite courses and the capstone course. The capstone course (SDP) shows a mean score above 85, indicating that most students were able to complete the project successfully, although with noticeable variation in performance.

The presence of sufficient variability across variables, as shown in Table 1, supports the suitability of the data for regression analysis, as it allows meaningful relationships between predictors and capstone performance to be examined.

Table 1. Descriptive Statistics of All Variables

Variabel	Mean	Std. Dev	Min	Max
GPA	3.44	0.28	2.5	4.0
BP	73.96	12.1	40	100
DS	65.31	15.3	30	100
DB	82.71	10.9	50	100
SWE	87.82	8.5	60	100
OOP	82.80	12.0	50	100
SPM	83.50	10.4	40	100
PBP	84.64	9.2	50	100
WP	83.50	11.2	40	100
SDP	85.72	9.8	50	100

4.2. Correlation Analysis

To explore the initial relationships between prerequisite courses and capstone performance, Pearson correlation analysis was conducted. The correlation matrix is visualized using a heatmap in Figure 1, which provides an overview of the strength and direction of correlations among all variables.

As shown in Figure 1, most programming-related courses exhibit moderate to strong positive correlations with SDP. The strongest correlations with capstone performance are observed for Object-Oriented Programming (OOP), Data Structures (DS), and Web Programming (WP). Database Systems (DB) also shows a positive simple correlation with SDP, suggesting that database knowledge is generally related to capstone outcomes when examined independently.

In addition, Figure 1 reveals relatively high correlations among several prerequisite courses, indicating potential overlap in the skills assessed by these courses. This observation motivates the use of multivariate regression analysis to identify the unique contribution of each course.

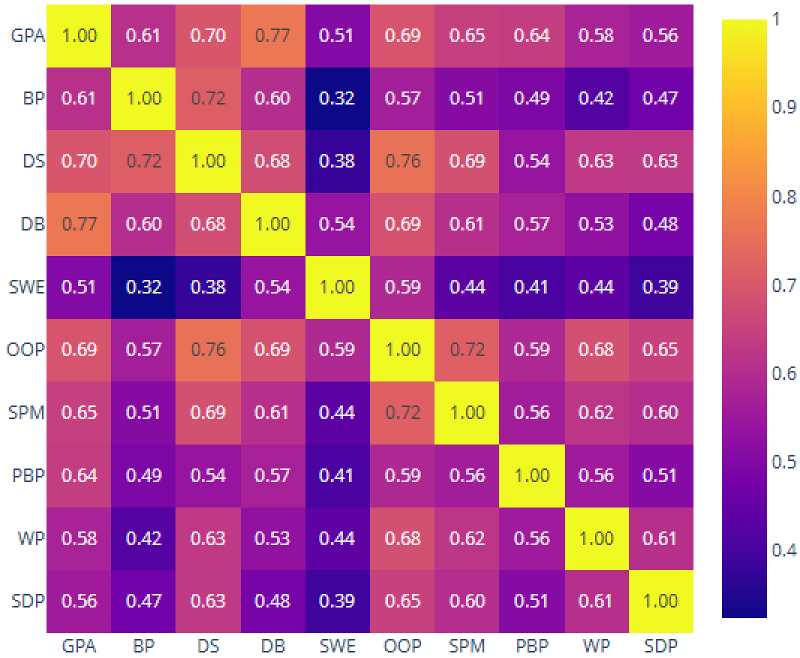

Figure 1. Correlation Heatmap

4.3. Multiple Linear Regression Results (OLS)

Table 2 presents the OLS regression results, including regression coefficients, standard errors, and statistical significance values. The overall model is statistically significant and achieves an R^2 value above 0.50, indicating that more than half of the variance in SDP can be explained by grades in prerequisite courses and GPA. As shown in Table 2, DS, OOP, SPM, PBP, and WP have statistically significant positive coefficients, indicating that strong performance in these courses is associated with higher capstone scores after controlling for other predictors.

To facilitate comparison across variables, standardized regression coefficients are shown in Figure 2. This figure highlights DS, OOP, and WP as the predictors with the largest standardized effects on SDP. In contrast, Basic Programming (BP) and Software Engineering (SWE) do not show statistically significant direct effects in the full model. GPA, included as a control variable, also shows a limited effect once course-level performance is taken into account.

Table 2. OLS Regression Results

Variable	Coefisien	Std. Error	t-Stat	p-Value	Significance
Const	-74.1806	13.855	-5.354	0.000	***
GPA	2.800	1.990	1.410	0.161	
BP	0.0833	0.0677	1.231	0.219	
DS	0.2189	0.0884	2.475	0.014	**
DB	-0.2001	0.107	-1.870	0.062	
SWE	0.2098	0.1685	1.245	0.214	
OOP	0.4294	0.1110	3.867	0.000	***
SPM	0.4311	0.1190	3.621	0.000	***
PBP	0.2006	0.0924	2.169	0.031	**
WP	0.3630	0.0930	3.913	0.000	***

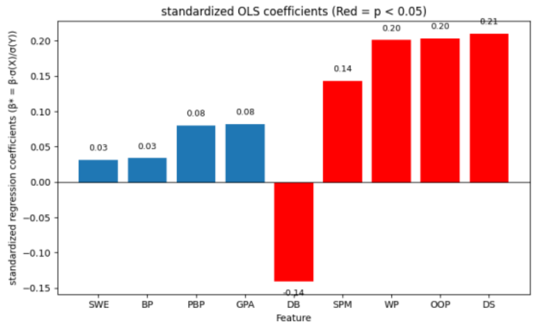

Figure 2. Standardized OLS Coefficients

4.4. Partial Correlation Analysis

To further examine the direct relationship between each prerequisite course and capstone performance, partial correlation analysis was conducted while controlling for all other predictors. The results are summarized in Table 3, which compares simple correlations and partial correlations for each course.

As shown in Table 3, DS, OOP, SPM, PBP, and WP retain positive and statistically significant partial correlations with SDP. This indicates that these courses have a direct relationship with capstone performance that is not fully explained by other courses.

In contrast, BP and SWE lose statistical significance after controlling for other variables, suggesting that their influence on capstone performance is largely indirect. A notable finding is observed for Database Systems (DB), which shows a change from a positive simple correlation to a negative partial correlation. This pattern indicates a suppression effect, where the apparent positive relationship between DB and SDP is primarily due to shared variance with other programming courses.

The difference between simple and partial correlations is illustrated in Figure 3, which highlights how the strength and direction of relationships change after controlling for overlapping predictors. This visualization helps clarify why certain courses remain influential while others do not. In Figure 3, DS, OOP, SPM, PBP, and WP remain positive after controlling for other predictors, although their values are lower than their simple correlations. This means that these courses still have a direct relationship with capstone performance. BP and SWE become very weak after control variables are included, suggesting that their effects are mainly indirect. DB shows the clearest change because its relationship changes from positive to negative, indicating a suppression effect.

Table 3. Simple vs. Partial Correlations

Variable	Simple Correlation (r)	p-value (Simple Correlation)	Partial Correlation (r)	p-value (Partial Correlation)	Direction Change?
BP	0.395	<0.001	0.052	0.278	No
DS	0.631	<0.001	0.285	<0.001	No
DB	0.481	<0.001	-0.116	0.0115	Yes
SWE	0.412	<0.001	0.046	0.329	No
OOP	0.648	<0.001	0.309	<0.001	No
SPM	0.602	<0.001	0.252	<0.001	No
PBP	0.505	<0.001	0.142	0.002	No
WP	0.653	<0.001	0.291	<0.001	No

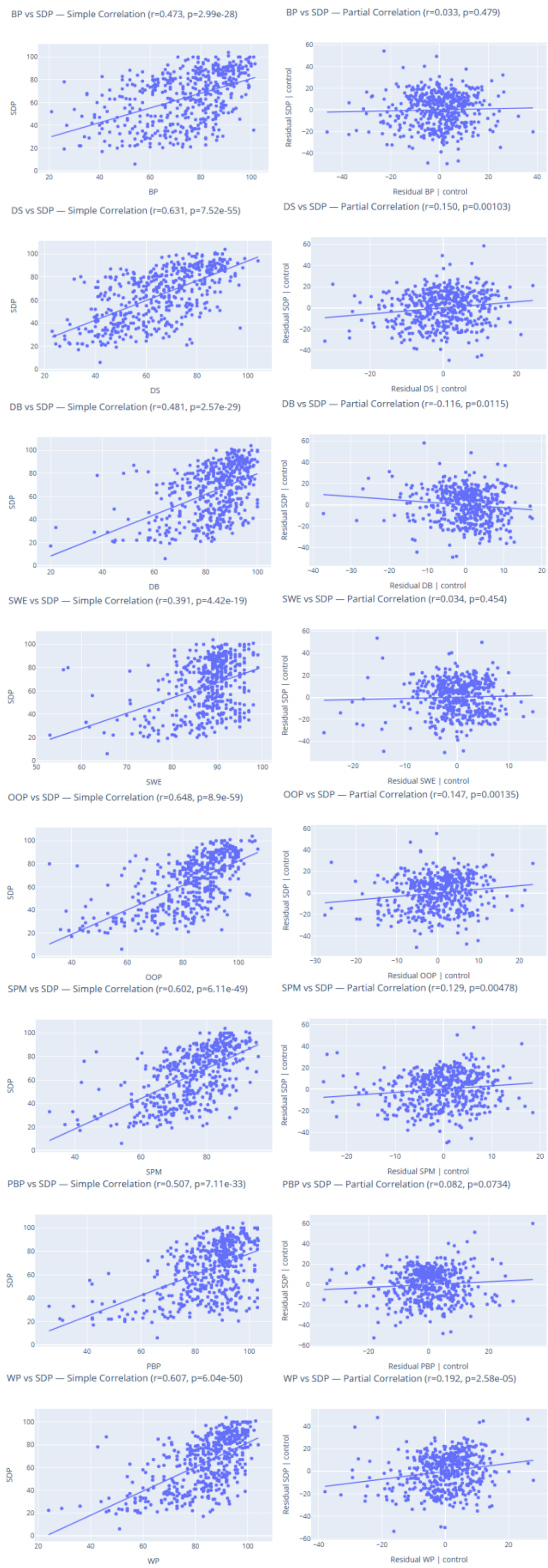

Figure 3. Comparison of Simple and Partial Correlations

4.5. LASSO Regression Results

LASSO Regression was applied as a supporting analysis to examine the stability of predictors identified by OLS and to reduce potential redundancy among variables. The standardized LASSO coefficients are shown in Figure 4. Figure 4 indicates that DS, OOP, WP, and SPM remain the most influential predictors after regularization. Variables with weaker effects, such as BP and SWE, are strongly shrunk toward zero. Database Systems (DB) again shows a reduced or negative coefficient, consistent with the findings from OLS and partial correlation analysis. The consistency between OLS and LASSO results suggests that the main findings are robust and not driven by multicollinearity or overfitting.

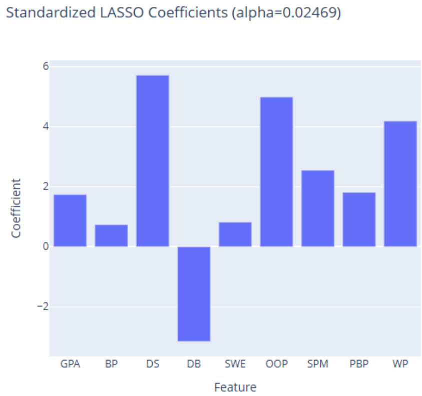

Figure 4. LASSO Coefficient Plot

4.6. Model Evaluation on Test Data

To evaluate predictive performance, an 80:20 train–test split was applied. Model performance was assessed using R^2, RMSE, and MAE. The results are summarized in Table 3, which compares the performance of OLS, LASSO, and a baseline model that predicts the mean SDP score.

As shown in Table 4, both OLS and LASSO models achieve R^2 values of approximately 0.42 on the test set, indicating that around 42% of the variation in unseen capstone performance can be predicted using prerequisite course grades. RMSE and MAE values indicate moderate prediction error, which is acceptable in educational prediction tasks. The baseline model yields a negative R^2, confirming that the regression models provide meaningful predictive value beyond naive estimation.

Table 4. Model Performance on Test Data

Model	R^2_test	RMSE_test	MAE_test
Baseline (mean)	-0.0097	20.8055	17.8269
OLS	0.4233	15.7239	12.1878
LASSO (α=0.02469)	0.4235	15.7214	12.1950

5. Discussion and Implications

5.1. Interpretation of Key Findings

This study examined the contribution of individual programming and database courses to student performance in a software engineering capstone course using Multiple Linear Regression as the main analytical model. The results presented in Table 2 and visualized in Figure 2 show that not all prerequisite courses contribute equally to capstone performance.

Courses such as Data Structures (DS), Object-Oriented Programming (OOP), and Web Programming (WP) emerge as the strongest predictors of capstone success. These findings are consistent with the correlation patterns shown in Figure 1, where these courses also exhibit strong positive relationships with SDP. From a practical perspective, these courses emphasize problem decomposition, abstraction, and system implementation, which are directly aligned with the technical demands of capstone projects.

In contrast, introductory or broadly scoped courses such as Basic Programming (BP) and Software Engineering (SWE) do not show a significant direct effect in the full regression model (Table 2). As indicated by the partial correlation results in Table 3, the influence of these courses appears to be indirect, operating through more advanced programming courses. This suggests that their role is foundational rather than immediately predictive of capstone outcomes.

5.2. The Role of Database Systems in Capstone Performance

One of the most notable findings of this study concerns the role of the Database Systems (DB) course. While DB shows a positive simple correlation with capstone performance (Figure 1), its unique contribution becomes negative after controlling for other courses, as shown by the partial correlation analysis in Table 3 and illustrated in Figure 3.

This suppression effect indicates that database knowledge alone does not independently predict capstone success when students already possess strong programming and system development skills. Instead, database concepts may function as supporting knowledge that is most effective when integrated into broader software development activities.

This result aligns with prior work highlighting that database skills are often embedded within broader system development activities, and that their impact becomes clearer when consistently reinforced in applied and project-based learning contexts[6]. The present findings suggest that database concepts may be more effective when consistently reinforced within applied programming and project-based courses, rather than being treated as isolated technical components.

5.3. Robustness of the Regression Findings

The consistency between the OLS results (Table 2), partial correlation analysis (Table 3), and LASSO regression (Figure 4) strengthens confidence in the robustness of the findings. LASSO regression confirms that the most influential predictors (DS, OOP, WP, and SPM) remain important even after regularization, while less influential variables are systematically shrunk toward zero.

Furthermore, the predictive evaluation reported in Table 3 shows that both OLS and LASSO models achieve similar performance on unseen data, with R^2 values around 0.42. This indicates that the identified relationships are not limited to the training data and have practical predictive value, even though a substantial portion of capstone performance is still influenced by non-academic factors. This finding is consistent with prior studies showing that regression-based models can provide reliable performance when academic variables are well structured and interpretable[14].

5.4. Implications for Curriculum Design

The findings of this study provide several important implications for curriculum design in software engineering and informatics programs. First, the strong and consistent influence of DS, OOP, and WP suggests that these courses play a central role in preparing students for capstone projects. As shown in Figure 2, these courses have the largest standardized effects on SDP, indicating that improvements in these areas are likely to have a direct impact on capstone outcomes. Institutions may consider strengthening learning outcomes, assessment strategies, and instructional support in these courses.

Second, the indirect role of introductory courses such as BP and SWE suggests that their effectiveness should be evaluated in terms of how well they prepare students for advanced coursework, rather than their direct association with capstone performance. Alignment between learning objectives across course sequences may help ensure that foundational knowledge is successfully transferred to later stages of the curriculum.

Third, the findings related to DB suggest that database skills, while already taught through practical activities such as ERD design and SQL implementation, may have a stronger impact on capstone performance when they are consistently reinforced within integrated software development contexts. Rather than altering the core content of database courses, curriculum designers may consider strengthening the continuity and reuse of database concepts in project-based programming courses and capstone preparation activities.

5.5. Implications for Academic Monitoring and Early Intervention

Beyond curriculum design, the results also suggest practical implications for academic monitoring and early intervention. The regression model presented in this study can be used as a screening tool to identify students who may face challenges in capstone courses based on their performance in key prerequisite subjects.

As indicated by the predictive performance in Table 3, while the model does not capture all factors influencing capstone success, it provides meaningful information at the cohort level. Academic advisors and program coordinators may use insights from key predictor courses (such as OOP and DS) to offer targeted support or preparatory interventions before students enter the capstone phase.

5.6. Summary of Discussion
In summary, the discussion of results shows that capstone performance is strongly influenced by specific advanced programming and project-oriented courses, while the role of database courses is more context-dependent and indirect. By linking regression results (Table 2), correlation analyses (Figures 1 and 3), and robustness checks (Figure 4, Table 3), this study provides a coherent and data-driven interpretation of how prerequisite courses contribute to capstone success.

These findings support a shift from broad academic indicators toward course-level analysis in curriculum evaluation, offering practical insights for improving student preparation and capstone outcomes.

6. Conclusion and Future Work
6.1. Conclusion
This study examined the contribution of individual programming and database courses to student performance in a software engineering capstone course using Multiple Linear Regression. By relying on course-level academic data rather than aggregate indicators, the analysis provides an interpretable view of how prerequisite courses relate to capstone outcomes.

The results indicate that advanced and applied courses (particularly Data Structures, Object-Oriented Programming, Web Programming, Software Project Management, and Platform-Based Programming) play an important role in preparing students for capstone projects. In contrast, introductory courses show mainly indirect effects, while the contribution of Database Systems appears context-dependent and most effective when reinforced within applied development activities.

The regression models achieve moderate predictive performance, highlighting both the value of academic course data and the influence of non-academic factors. Overall, the findings demonstrate that course-level regression analysis can support informed curriculum evaluation and academic decision-making in software engineering education.

6.2. Limitations
Several limitations of this study should be acknowledged. First, the analysis relies solely on academic grade data, which may not fully capture important factors such as teamwork, motivation, or prior project experience. Second, the dataset originates from a single academic program, which may limit the generalizability of the findings to other institutions with different curricula or grading practices. In addition, although regression models offer interpretability, they assume linear relationships between predictors and outcomes. More complex patterns may exist but are not explored within the scope of this study.

6.3. Future Work
Future research may extend this study by examining additional academic indicators that are readily available in institutional records, such as course repetition, grade trends, or variations in assessment components. Further analysis could also explore alternative model specifications using the same dataset, for example by adjusting feature selection strategies or control variables, to examine the stability of the findings. In addition, future studies may replicate the analysis using data from different academic cohorts within the same institution to assess the consistency of the observed relationships over time.

7. References
[1] M. C. Bastarrica, D. Perovich, and M. M. Samary, “What can students get from a software engineering capstone course?,” in *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering and Education Track, ICSE-SEET 2017*,

Institute of Electrical and Electronics Engineers Inc., Jun. 2017, pp. 137–145. doi: 10.1109/ICSE-SEET.2017.15.

[2] S. Tenhunen, T. Männistö, M. Luukkainen, and P. Ihantola, "A systematic literature review of capstone courses in software engineering," *Inf Softw Technol*, vol. 159, p. 1, Jul. 2023, doi: 10.1016/j.infsof.2023.107191.

[3] S. Howe, L. Rosenbauer, and S. Poulos, "The 2015 Capstone Design Survey Results: Current Practices and Changes over Time," *International Journal of Engineering Education*, vol. 33, no. 5, pp. 1393–1421, 2017, [Online]. Available: https://scholarworks.smith.edu/egr_facpubs/9

[4] R. P. Bringula, A. D.V. Aviles, Ma. Ymelda C. Batalla, Ma. Teresa F. Borebor, M. Anthony D. Uy, and B. E. San Diego, "Factors Affecting Failing the Programming Skill Examination of Computing Students," *International Journal of Modern Education and Computer Science*, vol. 9, no. 5, pp. 1–8, May 2017, doi: 10.5815/ijmecs.2017.05.01.

[5] S. Valstar, W. G. Griswold, and L. Porter, "The relationship between prerequisite proficiency and student performance in an upper-division computing course," in *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, Association for Computing Machinery, Inc, Feb. 2019, pp. 794–800. doi: 10.1145/3287324.3287419.

[6] E. D. Canedo, I. N. Bandeira, and P. H. T. Costa, "Challenges of Database Systems Teaching Amidst the Covid-19 Pandemic," in *Frontiers in Education Conference, FIE*, Lincoln, NE, USA: Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1–9. doi: 10.1109/FIE49875.2021.9637223.

[7] O. D. Oyerinde and P. A. Chia, "Predicting Students' Academic Performances-A Learning Analytics Approach using Multiple Linear Regression," *Int J Comput Appl*, vol. 157, no. 4, pp. 37–44, Jan. 2017.

[8] S. J. H. Yang, O. H. T. Lu, A. Y. Q. Huang, J. C. H. Huang, H. Ogata, and A. J. Q. Lin, "Predicting students' academic performance using multiple linear regression and principal component analysis," *Journal of Information Processing*, vol. 26, pp. 170–176, Jan. 2018, doi: 10.2197/ipsjjip.26.170.

[9] O. T. Omolewa, A. T. Oladele, A. A. Adeyinka, and O. R. Oluwaseun, "Prediction of student's academic performance using k-means clustering and multiple linear regressions," *Journal of Engineering and Applied Sciences*, vol. 14, no. 22, pp. 8254–8260, 2019.

[10] S. M. Khan, S. Muhammad, and S. A. Haider, "Performance Prediction of Computer Science Students in Capstone Software Engineering Course Through Educational Data Mining," in *2021 ASEE Virtual Annual Conference Content Access*, Arkansas Tech University, Jul. 2021. [Online]. Available: https://orc.library.atu.edu/faculty_pub_curr

[11] S. A. Licorish, M. Galster, G. M. Kapitsaki, and A. Tahir, "Understanding students' software development projects: Effort, performance, satisfaction, skills and their relation to the adequacy of outcomes developed," *Journal of Systems and Software*, vol. 186, pp. 1–0, Apr. 2022, doi: 10.1016/j.jss.2021.111156.

[12] A. M. Shahiri, W. Husain, and N. A. Rashid, "A Review on Predicting Student's Performance Using Data Mining Techniques," *Procedia Comput Sci*, vol. 72, pp. 414–422, 2015, doi: 10.1016/j.procs.2015.12.157.

[13] P. Dabhade, R. Agarwal, K. P. Alameen, A. T. Fathima, R. Sridharan, and G. Gopakumar, "Educational data mining for predicting students' academic performance using machine learning algorithms," in *Materials Today: Proceedings*, Elsevier Ltd, 2021, pp. 5260–5267. doi: 10.1016/j.matpr.2021.05.646.

[14] A. A. Saa, "Educational Data Mining & Students' Performance Prediction," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, pp. 212–220, 2016, [Online]. Available: www.ijacsa.thesai.org

[15] M.-S. Kim, Y.-H. Jeong, and G.-W. Kwak, "Factors affecting the performance of student project teams in capstone design programmes," *Global Journal of Engineering Education*, vol. 26, no. 1, pp. 13–19, 2024.