Comprehensive Grading Criteria for Computer Science Assignments

1. Introduction: Establishing Comprehensive Grading Criteria for Computer Science Assignments

The evaluation of student code stands as a cornerstone of Computer Science education, playing a vital role in assessing learning, providing feedback, and ensuring the development of essential programming skills. However, the diverse nature of Computer Science, encompassing various levels of complexity and a multitude of specialized domains, presents significant challenges in creating standardized yet meaningful evaluation processes. The spectrum of assignments, ranging from introductory exercises to advanced projects in areas like data structures, machine learning, and web development, necessitates a flexible framework that can effectively gauge student proficiency across this broad landscape. [1]

This document addresses the need for a coherent and detailed approach to grading by presenting a comprehensive rubric designed for evaluating student code in Computer Science assignments. The rubric is structured to accommodate beginner, intermediate, and advanced skill levels and is applicable across different domains within the field. Its development involved a thorough synthesis of various grading rubrics and best practices found in online educational resources, focusing on providing clear, qualitative ratings of 'Poor', 'Fair', and 'Excellent' instead of traditional numerical score ranges. This shift towards qualitative feedback aims to offer students more descriptive and actionable insights into their performance, fostering a deeper understanding of the criteria for success. [2]

Recognizing the diverse educational environments in which Computer Science is taught, this grading criteria document is designed to be adaptable for both traditional university programs and intensive coding bootcamps. The principles and criteria outlined herein can be adjusted to align with the specific learning objectives, curriculum focus, and assessment styles prevalent in these different educational settings. By providing a flexible yet robust framework, this rubric seeks to enhance the consistency and fairness of code evaluation, ultimately contributing to a more effective learning experience for students across all levels of Computer Science education. [3]

2. Beginner Level Coding Assignment Rubric

Criterion	Poor	Fair	Excellent
Implementation Correctness	Code fails to execute or produces incorrect output for most test cases. Significant logical errors are present.	Code executes with some errors or produces partially correct output. Basic logical errors may be present. May pass only a few basic test cases. [5]	Code executes correctly and produces the expected output for all or nearly all test cases. Demonstrates a solid grasp of fundamental programming logic. [5]
Understanding of Fundamental Concepts	Demonstrates a lack of understanding of basic programming concepts such as variables, data types, control flow (if/else, loops), and basic syntax. [6]	Shows a partial understanding of fundamental concepts but may misuse or incorrectly apply them in the code.	Clearly demonstrates a strong understanding of fundamental programming concepts and applies them correctly and appropriately in the code. [6]
Code Readability and Style	Code is poorly formatted, difficult to read, lacks comments, and uses inconsistent indentation. Variable names are unclear or misleading. [7]	Code formatting is inconsistent, readability is somewhat compromised, comments are sparse or unclear, and indentation may have issues. Variable names could be more descriptive. [8]	Code is well-formatted, easy to read, includes clear and concise comments explaining the logic, and uses consistent indentation. Variable names are descriptive and follow basic naming conventions. [8]
Basic HTML and CSS Structure (if applicable)	HTML structure is invalid or poorly organized. Basic elements are misused or missing. CSS is absent or completely ineffective. [9]	HTML structure is mostly valid but may have some organizational issues. Basic elements are generally used correctly. CSS is present but may have significant errors or	HTML structure is valid, well-organized, and uses semantic elements appropriately. Basic elements are used correctly to structure the content. CSS is used effectively to

		inconsistencies in styling. [9]	style the page with consistent and appropriate styling. [9]
Basic Program Functionality	Program lacks essential functionality as described in the assignment requirements. Core features are missing or non-functional.	Program implements some of the core functionality but may be missing key features or have incomplete implementations.	Program fully implements all essential functionality as described in the assignment requirements. Basic features work as expected. [9]
Fundamental Python Usage (Syntax, Data Types)	Demonstrates significant errors in Python syntax and a lack of understanding of basic data types (integers, floats, strings, lists, dictionaries). [6]	Shows some understanding of Python syntax and basic data types but may have inconsistencies or errors in their usage.	Uses Python syntax correctly and demonstrates a solid understanding of basic data types, applying them appropriately in the code. [6]
Introduction to Basic Framework Usage (e.g., simple React components, basic Flask application)	Fails to use the specified framework or demonstrates a complete misunderstanding of its basic concepts (e.g., cannot create a basic React component or a simple Flask route). [13]	Attempts to use the specified framework but shows significant errors or limitations in applying its basic concepts (e.g., creates a React component with errors or a Flask application that doesn't run). [13]	Correctly uses the basic concepts of the specified framework (e.g., creates functional simple React components or a basic Flask application with routes). [13]

3. Intermediate Level Coding Assignment Rubric

Criterion	**Poor**	**Fair**	**Excellent**
Efficient Implementation and Algorithm Design	Implementation is inefficient, uses brute-force approaches where	Implementation shows some awareness of efficiency but may	Implementation is efficient, utilizes appropriate algorithms for the

	more efficient algorithms exist, and may have significant performance issues. Algorithm design is flawed or poorly conceived. [16]	still contain unnecessary steps or suboptimal algorithms. Algorithm design is adequate but could be improved for efficiency. [16]	task, and demonstrates consideration for performance. Algorithm design is well-thought-out and optimized for the problem. [16]
Appropriate Selection and Use of Data Structures	Uses inappropriate data structures for the task, leading to inefficient operations or difficulties in implementation. Demonstrates a lack of understanding of the properties of different data structures. [16]	Selects generally appropriate data structures but may not utilize them effectively or may miss opportunities for optimization through better data structure choices. Shows a basic understanding of common data structures. [16]	Selects and uses data structures that are highly appropriate for the task, demonstrating a strong understanding of their properties and how they impact efficiency. [16]
Understanding and Application of Time and Space Complexity	Shows little to no understanding of time and space complexity. Code may have significant performance bottlenecks due to inefficient algorithms or data structure usage without awareness of the complexity implications. [18]	Demonstrates a basic understanding of time and space complexity but may not be able to analyze their own code effectively or apply this knowledge to optimize their solutions. [18]	Clearly understands and can analyze the time and space complexity of their code. Applies this knowledge to design efficient algorithms and choose appropriate data structures. May include documentation of the complexity analysis. [18]
Effective Data Preprocessing Techniques (Python, R, MATLAB)	Fails to preprocess data appropriately, leading to issues with model training or analysis (e.g., missing values not handled, data not scaled).	Attempts to preprocess data but may miss important steps or apply techniques incorrectly. Shows a basic understanding	Effectively applies appropriate data preprocessing techniques relevant to the task and the chosen language/tools.

	Demonstrates a lack of understanding of common preprocessing steps. [21]	of some common preprocessing techniques. [21]	Demonstrates a strong understanding of why and how to preprocess data for better model performance or analysis. [21]
Suitable Model Selection and Application (Python, R, MATLAB, TensorFlow, scikit-learn)	Selects an inappropriate model for the task or applies it incorrectly. Demonstrates a lack of understanding of the strengths and weaknesses of different models. [26]	Selects a generally suitable model but may not apply it optimally or may not justify the choice effectively. Shows a basic understanding of some common models. [26]	Selects a highly suitable model for the task, applies it correctly, and provides a clear justification for the choice based on the problem requirements and data characteristics. [26]
Application of Basic Evaluation Metrics (Python, R, MATLAB)	Fails to evaluate the model's performance or uses inappropriate metrics. Demonstrates a lack of understanding of basic evaluation metrics (e.g., accuracy, precision, recall, F1-score for classification; RMSE, MAE, R-squared for regression). [30]	Applies some basic evaluation metrics but may not interpret them correctly or may not choose the most relevant metrics for the task. Shows a basic understanding of common evaluation metrics. [30]	Applies appropriate basic evaluation metrics to assess the model's performance and interprets the results correctly. Justifies the choice of metrics based on the problem context. [30]
Competent Use of Relevant Frameworks (React, Angular, Vue.js, Django, Flask, ASP.NET, TensorFlow, PyTorch, scikit-learn)	Demonstrates significant errors or misunderstandings in using the specified framework. Fails to utilize key features or follows basic conventions. [14]	Uses the specified framework but may have inconsistencies, inefficiencies, or miss opportunities to leverage more advanced features. Shows a basic understanding of the framework's structure and	Uses the specified framework competently, applying its key features and following best practices and conventions. Demonstrates a good understanding of the framework's

		conventions. [14]	architecture and capabilities. [14]
Implementation of Server-Side Logic (Python/Flask/Django, ASP.NET)	Fails to implement server-side logic or demonstrates significant errors in its implementation. Shows a lack of understanding of basic server-side concepts (e.g., routing, request handling). [51]	Attempts to implement server-side logic but may have incomplete or incorrect implementations. Shows a basic understanding of some server-side concepts. [51]	Implements server-side logic correctly and effectively, demonstrating a good understanding of relevant concepts such as routing, request handling, and basic security considerations. [51]
Basic Database Interaction	Fails to interact with a database or demonstrates significant errors in basic database operations (e.g., connecting, querying, inserting data). Shows a lack of understanding of database fundamentals. [56]	Attempts to interact with a database but may have inconsistencies or errors in basic operations. Shows a basic understanding of some database fundamentals. [56]	Interacts with a database correctly and effectively, performing basic operations such as connecting, querying, inserting, and updating data as required by the assignment. Demonstrates a good understanding of database fundamentals. [56]
Appropriate and Effective Use of Language-Specific Features (Python, React, R, MATLAB)	Misuses or fails to leverage important language-specific features or libraries relevant to the assignment. Code may be verbose or inefficient due to a lack of familiarity with idiomatic language use. [6]	Uses some language-specific features but may not apply them effectively or consistently. Code could be more concise or idiomatic. [6]	Demonstrates appropriate and effective use of language-specific features, libraries, and idiomatic constructs, resulting in clean, concise, and efficient code. [6]

4. Advanced Level Coding Assignment Rubric

Criterion	Poor	Fair	Excellent
Innovation and Problem-Solving	Project lacks originality and does not demonstrate innovative problem-solving. The solution is a straightforward implementation of known concepts without any creative extensions or novel approaches. [67]	Project shows some attempts at originality or innovation but may not be fully realized or impactful. Problem-solving demonstrates a basic understanding but lacks sophistication. [67]	Project demonstrates significant innovation, originality, and creative problem-solving. The solution goes beyond standard approaches and may introduce novel techniques or insightful extensions. [67]
Code Efficiency and Performance Optimization	Code is highly inefficient, may have significant performance bottlenecks, and does not demonstrate any consideration for optimization. [86]	Code shows some consideration for efficiency but may still have areas for improvement. Basic optimization techniques may be attempted but not fully implemented or effective. [86]	Code is highly efficient, demonstrates a deep understanding of performance considerations, and utilizes advanced optimization techniques where appropriate. [86]
Scalability and Robustness of Solutions	Solution is not scalable and would likely fail or perform poorly with increased load or data volume. Error handling is inadequate or absent. [64]	Solution demonstrates some consideration for scalability but may have limitations or vulnerabilities under heavy load. Basic error handling may be implemented. [64]	Solution is highly scalable and robust, capable of handling significant increases in load and data volume. Implements comprehensive error handling and demonstrates resilience to failures. [64]
Application of Sophisticated Techniques in Data Structures and	Relies only on basic data structures and algorithms. Fails to utilize more advanced techniques	Attempts to use more sophisticated techniques but may not apply them correctly or	Effectively applies sophisticated techniques in data structures and algorithms relevant to

Algorithms	appropriate for the problem. [106]	effectively. Shows a basic understanding of advanced data structures and algorithms. [106]	the problem, demonstrating a deep understanding of their principles and trade-offs. [106]
Advanced Machine Learning Techniques and Framework Utilization (TensorFlow, PyTorch)	Fails to utilize advanced ML techniques or demonstrates a significant lack of understanding of the chosen framework (TensorFlow or PyTorch). [1]	Attempts to use advanced ML techniques or the chosen framework but may have limitations, errors, or inefficiencies. Shows a basic understanding of some advanced concepts and framework functionalities. [1]	Effectively utilizes advanced ML techniques and demonstrates a strong understanding of the chosen framework (TensorFlow or PyTorch), applying best practices for model development, training, and evaluation. [1]
Mastery of Web Development Frameworks and Best Practices (React, Angular, Vue.js, Django, Flask, ASP.NET)	Demonstrates a weak understanding of advanced features and best practices in the chosen web development framework. Project may have significant architectural flaws or security vulnerabilities. [13]	Shows some understanding of advanced features and best practices but may not apply them consistently or effectively. Project architecture may have some limitations or areas for improvement. [13]	Demonstrates mastery of advanced features and adheres to best practices in the chosen web development framework. Project architecture is well-designed, scalable, and secure. [13]
Advanced Features and Best Practices in Domain-Specific Languages (Python, React, R, MATLAB)	Fails to utilize advanced features or follow best practices specific to the programming languages used in the project. Code may be inefficient, error-prone, or not idiomatic. [6]	Uses some advanced language-specific features but may not apply them effectively or consistently. Code could be improved by adhering more closely to best practices. [6]	Demonstrates mastery of advanced features and consistently follows best practices specific to the programming languages used, resulting in high-quality, efficient, and idiomatic code. [6]

5. Conclusion: Applying the Comprehensive Grading Criteria

The grading criteria outlined in this document provide a comprehensive framework for evaluating student code across various levels and domains within Computer Science education. By synthesizing insights from a wide range of online resources and focusing on qualitative ratings, this rubric aims to offer educators a robust tool for assessing student work in a consistent and meaningful manner. The criteria are designed to evaluate not only the correctness of the code but also the student's understanding of fundamental and advanced concepts, their ability to write efficient and scalable solutions, and their proficiency in using relevant programming languages and frameworks.

The adaptability of this rubric makes it suitable for diverse educational contexts, including university programs that often emphasize theoretical foundations and coding bootcamps that typically focus on practical, job-ready skills. Instructors in both settings can leverage these criteria to align their evaluations with specific learning objectives and assignment requirements. The detailed descriptions for 'Poor', 'Fair', and 'Excellent' ratings provide clear benchmarks for student performance, enabling educators to offer targeted feedback that supports student growth and development. [2]

By adopting a standardized rubric like the one presented here, educators can ensure a greater degree of consistency and fairness in their grading practices. This not only benefits students by providing them with a clear understanding of expectations but also streamlines the evaluation process for instructors, allowing them to focus on providing constructive feedback that enhances the learning experience. The comprehensive nature of these criteria supports a holistic assessment of student coding abilities, preparing them for future challenges in their academic and professional careers.

Works cited

1. A Rubric for Evaluating New Projects that Produce Data - Analytic Strategy Partners, accessed on May 1, 2025, https://analyticstrategy.com/a-rubric-for-evaluating-new-projects-that-produce-data/
2. Rubric Handbook - Florida Memorial University, accessed on May 1, 2025, https://www.fmu.edu/wp-content/uploads/2021/05/Rubric_Handbook.pdf
3. What Is a Coding Bootcamp? Requirements and Steps - Fullstack Academy, accessed on May 1, 2025, https://www.fullstackacademy.com/blog/what-is-a-coding-bootcamp
4. Coding Bootcamps in 2025: Your Complete Guide - Course Report, accessed on

May 1, 2025, https://www.coursereport.com/coding-bootcamp-ultimate-guide
5. Create your first Python assignment - CodeGrade Help, accessed on May 1, 2025, https://help.codegrade.com/automatic-grading-guides/python/create-your-first-python-assignment
6. 10-Objects and Data Structures Assessment Test-Solution.ipynb - GitHub, accessed on May 1, 2025, https://github.com/Pierian-Data/Complete-Python-3-Bootcamp/blob/master/00-Python%20Object%20and%20Data%20Structure%20Basics/10-Objects%20and%20Data%20Structures%20Assessment%20Test-Solution.ipynb
7. The Java "style" guide/rubric for my friend's data structures class - Reddit, accessed on May 1, 2025, https://www.reddit.com/r/programminghorror/comments/9tvmdt/the_java_style_guiderubric_for_my_friends_data/
8. Rubric for Computer Programming - LIU CS – Prof. League, accessed on May 1, 2025, https://liucs.net/cs102f15/programming-rubric.pdf
9. Web Site Project (Final Project / 30% of grade) - UDC, accessed on May 1, 2025, https://csit.udc.edu/~rpalomino/courses/summer11/wd/project/wd-final_project-guidelines.pdf
10. Project Guidelines, accessed on May 1, 2025, https://public.websites.umich.edu/~ece/resources/ece_project_guidelines.html
11. Project Rubric – Web Development - New Media Institute, accessed on May 1, 2025, https://nmi.cool/web/project-rubric/
12. 12 Web Dev Projects for Beginners & Intermediate - Teamwork.com, accessed on May 1, 2025, https://www.teamwork.com/blog/web-development-projects/
13. Unlock Web Magic: Build Your First Flask App (Beginner-Friendly Guide) - DEV Community, accessed on May 1, 2025, https://dev.to/alexrivers/unlock-web-magic-build-your-first-flask-app-beginner-friendly-guide-n1g
14. Flask Tutorial - GeeksforGeeks, accessed on May 1, 2025, https://www.geeksforgeeks.org/flask-tutorial/
15. Advanced PyTorch Techniques and Applications - Coursera, accessed on May 1, 2025, https://www.coursera.org/learn/packt-advanced-pytorch-techniques-and-applications-jmkne
16. 15 Data Structure and Algorithm (DSA) Project Ideas - Codegnan, accessed on May 1, 2025, https://codegnan.com/data-structure-and-algorithm-project-ideas/
17. Data Structures: A Comprehensive Introduction - DEV Community, accessed on May 1, 2025, https://dev.to/m__mdy__m/data-structures-a-comprehensive-introduction-2o13
18. Time Complexity Of Algorithms: Types, Notations, Cases, and More - Unstop, accessed on May 1, 2025, https://unstop.com/blog/time-complexity-of-algorithms
19. Complete Guide On Complexity Analysis – Data Structure and Algorithms Tutorial, accessed on May 1, 2025,

https://www.geeksforgeeks.org/complete-guide-on-complexity-analysis/
20. Big O Notation: Time Complexity & Examples Explained - Simplilearn.com, accessed on May 1, 2025, https://www.simplilearn.com/big-o-notation-in-data-structure-article
21. Data Preparation for Machine Learning: The Ultimate Guide to Doing It Right - Pecan AI, accessed on May 1, 2025, https://www.pecan.ai/blog/data-preparation-for-machine-learning/
22. Data Preprocessing and Feature Engineering in Machine Learning - Magnimind Academy, accessed on May 1, 2025, https://magnimindacademy.com/blog/data-preprocessing-and-feature-engineering-in-machine-learning/
23. A Comprehensive Guide to Data Preprocessing - Neptune.ai, accessed on May 1, 2025, https://neptune.ai/blog/data-preprocessing-guide
24. Data Preprocessing in Machine Learning [6 Best Practices] - Intelliarts, accessed on May 1, 2025, https://intelliarts.com/blog/data-preprocessing-in-machine-learning-best-practices/
25. What Is Data Preprocessing for Machine Learning? - Pure Storage, accessed on May 1, 2025, https://www.purestorage.com/knowledge/what-is-data-preprocessing.html
26. Probabilistic Model Selection with AIC, BIC, and MDL - MachineLearningMastery.com, accessed on May 1, 2025, https://machinelearningmastery.com/probabilistic-model-selection-measures/
27. Model Selection in Machine Learning - Applied AI Course, accessed on May 1, 2025, https://www.appliedaicourse.com/blog/model-selection-in-machine-learning/
28. A Gentle Introduction to Model Selection for Machine Learning - MachineLearningMastery.com, accessed on May 1, 2025, https://machinelearningmastery.com/a-gentle-introduction-to-model-selection-for-machine-learning/
29. Your 101 Guide to Model Selection In Machine Learning - ProjectPro, accessed on May 1, 2025, https://www.projectpro.io/article/model-selection-in-machine-learning/824
30. Evaluation Metrics: Easily Evaluate ML Model in a Project - Datasaur, accessed on May 1, 2025, https://datasaur.ai/blog-posts/ml-model-evaluation-metrics
31. Evaluation and monitoring metrics for generative AI - Azure AI Foundry | Microsoft Learn, accessed on May 1, 2025, https://learn.microsoft.com/en-us/azure/ai-foundry/concepts/evaluation-metrics-built-in
32. Magician's Corner: 9. Performance Metrics for Machine Learning Models - RSNA Journals, accessed on May 1, 2025, https://pubs.rsna.org/doi/full/10.1148/ryai.2021200126
33. 12 Important Model Evaluation Metrics for Machine Learning (2025) - Analytics Vidhya, accessed on May 1, 2025, https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-e

rror-metrics/
34. Overview of Evaluation Metrics for Classification Models | Keylabs, accessed on May 1, 2025, https://keylabs.ai/blog/overview-of-evaluation-metrics-for-classification-models/
35. Server-side web frameworks - Learn web development - MDN Web Docs, accessed on May 1, 2025, https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/First_steps/Web_frameworks
36. (PDF) Analyzing best practices on Web development frameworks: The lift approach, accessed on May 1, 2025, https://www.researchgate.net/publication/273187950_Analyzing_best_practices_on_Web_development_frameworks_The_lift_approach
37. Web Standards - W3C, accessed on May 1, 2025, https://www.w3.org/standards/
38. An Ultimate Guide to Web Application Framework in 2025 - Sencha.com, accessed on May 1, 2025, https://www.sencha.com/blog/a-comprehensive-guide-to-web-application-frameworks/
39. Top 10 Backend Frameworks [2024] - Daily.dev, accessed on May 1, 2025, https://daily.dev/blog/top-10-backend-frameworks-2024
40. The Best Web Development Frameworks in 2024 - DEV Community, accessed on May 1, 2025, https://dev.to/adityabhuyan/the-best-web-development-frameworks-in-2024-2bj3
41. Web Development Frameworks: A Guide | Built In, accessed on May 1, 2025, https://builtin.com/articles/web-development-frameworks
42. 8 Most Popular Web Programming Frameworks (2024) - App Academy, accessed on May 1, 2025, https://www.appacademy.io/blog/8-most-popular-web-programming-frameworks
43. Technology | 2024 Stack Overflow Developer Survey, accessed on May 1, 2025, https://survey.stackoverflow.co/2024/technology
44. Top Front-End Frameworks in 2024 Worth Your Time and Effort to Master - Refine dev, accessed on May 1, 2025, https://refine.dev/blog/best-front-end-frameworks-in-2024/
45. Top 10 Web Application Development Frameworks in 2025 - Sencha.com, accessed on May 1, 2025, https://www.sencha.com/blog/web-application-development-top-frameworks/
46. 32 Best Web Development Frameworks For 2025 - LambdaTest, accessed on May 1, 2025, https://www.lambdatest.com/blog/best-web-development-frameworks/
47. Django Tutorial | Learn Django Framework - GeeksforGeeks, accessed on May 1, 2025, https://www.geeksforgeeks.org/django-tutorial/
48. Top 50 Django Interview Questions and Answers | GeeksforGeeks, accessed on May 1, 2025, https://www.geeksforgeeks.org/django-interview-questions/
49. Spring Boot Interview Questions and Answers | GeeksforGeeks, accessed on May 1, 2025,

https://www.geeksforgeeks.org/spring-boot-interview-questions-and-answers/
50. Angular Tutorial for Beginners - Getting Started with Angular 8 | Edureka, accessed on May 1, 2025, https://www.edureka.co/blog/angular-tutorial/
51. Online Interactive Home Work Grading System, accessed on May 1, 2025, https://www.iiisci.org/journal/pdv/sci/pdfs/P792477.pdf
52. Top Server-side Scripting Languages & Frameworks - Orient Software, accessed on May 1, 2025, https://www.orientsoftware.com/blog/server-side-scripting-languages/
53. CS55.13: Server-Side Web Development Section 0508, Fall 2023 Course Syllabus - SRJC Login Portal - Santa Rosa Junior College, accessed on May 1, 2025, https://portal.santarosa.edu/SrWeb/SR_DisplayDocument.aspx?SectionDocumentID=12881
54. Advanced Internet Computing Systems and Applications | OMSCentral, accessed on May 1, 2025, https://www.omscentral.com/courses/advanced-internet-computing-systems-and-applications/reviews
55. Information Systems Management - ISMN < Auburn University, accessed on May 1, 2025, https://bulletin.auburn.edu/coursesofinstruction/ismn/
56. 15 Best Database for Web Applications and Development in 2025 - Synodus, accessed on May 1, 2025, https://synodus.com/blog/web-development/database-for-web-applications-and-development/
57. (PDF) Database design for course selection and course grading system - ResearchGate, accessed on May 1, 2025, https://www.researchgate.net/publication/378359008_Database_design_for_course_selection_and_course_grading_system
58. Classification of Software Requirements – Software Engineering - GeeksforGeeks, accessed on May 1, 2025, https://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/
59. Web Development and Digital Media (WEB) | Johnson County Community College Catalog, accessed on May 1, 2025, https://catalog.jccc.edu/coursedescriptions/web/
60. Real-time Databases: What developers need to know - Tinybird, accessed on May 1, 2025, https://www.tinybird.co/blog-posts/real-time-databases-what-developers-need-to-know
61. Grading Database Quality, Part 2: Database Models - Dataversity, accessed on May 1, 2025, https://www.dataversity.net/grading-database-quality-part-2-database-models/
62. Computer & Info. Tech. (CITP) < South Texas College, accessed on May 1, 2025, https://catalog.southtexascollege.edu/courses/citp/
63. Web Development I Proficiency Certificate | Community College of Philadelphia, accessed on May 1, 2025, https://www.ccp.edu/college-catalog/degree-programs/proficiency-certificate/w

eb-development-i-proficiency-certificate
64. Rubric Best Practices, Examples, and Templates - Teaching Resources, accessed on May 1, 2025, https://teaching-resources.delta.ncsu.edu/rubric_best-practices-examples-templates/
65. 8 Steps to Create a Good Rubric for Writing Assignments & 7 Digital Tools to Use, accessed on May 1, 2025, https://www.essaygrader.ai/blog/rubric-for-writing-assignments
66. MATLAB Rubrics | EF105, accessed on May 1, 2025, https://efcms.engr.utk.edu/ef105-2021-01/grav/resources/matlab-rubrics
67. Understanding, Grading, and Judging Computer Science Projects, accessed on May 1, 2025, https://www.sciencebuddies.org/science-fair-projects/ScienceBuddies_Grading_and_Judging_Computer_Science_Projects-draft.pdf
68. Initial Rubric Criteria Thoughts - CS in Science - Code.org Professional Learning Community, accessed on May 1, 2025, https://forum.code.org/t/initial-rubric-criteria-thoughts/2922
69. Grading Rubric – Creative Coding - Courses taught by Tega Brain, accessed on May 1, 2025, https://www.courses.tegabrain.com/cc18/grading-rubric/
70. How to write a grading rubric for coding assignments? - Computer Science Educators, accessed on May 1, 2025, https://cseducators.stackexchange.com/questions/5339/how-to-write-a-grading-rubric-for-coding-assignments
71. Coding Challenge Rubric | Nextech, accessed on May 1, 2025, https://nextech.org/wp-content/uploads/2020/04/Nextech-Coding-Competition-Rubric-1.pdf
72. Scratch Project Rubrics – Julian Thomas Screawn - JTS Design, accessed on May 1, 2025, https://jts.design/resources/scratch-programming-guide/scratch-project-rubrics/
73. Build Better Coding Assignment Rubrics - Tech Savvy Code, accessed on May 1, 2025, https://techsavvycode.com/2025/03/22/build-better-coding-assignment-rubrics/
74. Human vs. Automated Coding Style Grading in Computing Education, accessed on May 1, 2025, https://web.eecs.umich.edu/~weimerw/p/weimer-asee2019.pdf
75. Grading Rubrics - CSE 123 - Washington, accessed on May 1, 2025, https://courses.cs.washington.edu/courses/cse123/23au/rubrics/
76. 5 Ways to Grade Coding Lessons - Ellipsis Education, accessed on May 1, 2025, https://ellipsiseducation.com/blog/how-to-grade-computer-science-coding-lessons
77. Does anyone have a good rubric for evaluating student programming projects?, accessed on May 1, 2025, https://forum.code.org/t/does-anyone-have-a-good-rubric-for-evaluating-student-programming-projects/9902
78. Grading Rubric - CSE 122 - Washington, accessed on May 1, 2025, https://courses.cs.washington.edu/courses/cse122/23sp/grading_rubric/

79. How does everyone come up with ideas for projects? : r/learnprogramming - Reddit, accessed on May 1, 2025, https://www.reddit.com/r/learnprogramming/comments/opzrqa/how_does_everyone_come_up_with_ideas_for_projects/
80. Creative Coding Project Assessment Rubric | BrainPOP Educators, accessed on May 1, 2025, https://educators.brainpop.com/wp-content/uploads/2017/06/CreativeCodingRubric.docx.pdf
81. Validating a Creative Coding Rubric through expressive activities for elementary grades - Sites@BC, accessed on May 1, 2025, https://sites.bc.edu/devtech/wp-content/uploads/sites/181/2024/09/Validating-a-Creative-Coding-Rubric-through-expressive-activities-for-elementary-grades.pdf
82. ScratchJr Project Rubric – DevTech Research Group - Sites@BC, accessed on May 1, 2025, https://sites.bc.edu/devtech/assessments/scratchjr-instruments/scratchjr-project-rubric/
83. Rubric for the App Proposal Project - Unit and Lesson Discussion, accessed on May 1, 2025, https://forum.code.org/t/rubric-for-the-app-proposal-project/11920
84. Coding Project Rubric by The Creative Cause - TPT, accessed on May 1, 2025, https://www.teacherspayteachers.com/Product/Coding-Project-Rubric-10408074
85. Creative Coding Assessment Rubric - BrainPOP Educators, accessed on May 1, 2025, https://educators.brainpop.com/printable/creative-coding-assessment-rubric/
86. iRubric: Data Structure Lab rubric - BX32X83 - RCampus, accessed on May 1, 2025, https://www.rcampus.com/rubricshowc.cfm?code=BX32X83&sp=yes&
87. Rubric (Spring 2012) Senior Project Assessment of Student Outcomes of the BS in Computer Science of the School of Computing and, accessed on May 1, 2025, https://users.cs.fiu.edu/~pestaina/SU12_1_CloudControl_Rubric_PC.pdf
88. Rubric (Spring 2011) Senior Project Assessment of Student Outcomes of the BS in Computer Science of the School of Computing and, accessed on May 1, 2025, https://users.cs.fiu.edu/~pestaina/SP11_4_Geonomics_CheckList.pdf
89. Rubric (Spring 2013) Senior Project Assessment of Student Outcomes of the BS in Computer Science of the School of Computing and, accessed on May 1, 2025, https://users.cs.fiu.edu/~pestaina/SP13_1_Shout_Rubric_MS.pdf
90. Scalable Software Architecture – How to Grow Your Web Application Most Effectively, accessed on May 1, 2025, https://brainhub.eu/library/scalable-software-architecture
91. Best Practices for Designing Effective Rubrics | Teach Online - Arizona State University, accessed on May 1, 2025, https://teachonline.asu.edu/2019/02/best-practices-for-designing-effective-rubrics/
92. The design principles behind scalable web apps - Mattermost, accessed on May 1, 2025, https://mattermost.com/blog/design-principles-for-scalable-web-apps/

93. CTO's Guide to Scalable Web Application Development - Digital Aptech, accessed on May 1, 2025, https://digitalaptech.com/blogs/cto-guide-scalable-web-application-development/
94. How to make a scalable web application [architecture review] | JustCoded, accessed on May 1, 2025, https://justcoded.com/blog/how-to-make-a-successful-scalable-web-application/
95. Scale Your Web Application — One Step at a Time | AWS Architecture Blog, accessed on May 1, 2025, https://aws.amazon.com/blogs/architecture/scale-your-web-application-one-step-at-a-time/
96. How to Build a Scalable Web Application for Long-Term Success - Bitcot, accessed on May 1, 2025, https://www.bitcot.com/how-to-build-scalable-web-applications/
97. How to Build a Scalable Web Applications in 2025 : Aalpha, accessed on May 1, 2025, https://www.aalpha.net/blog/how-to-build-a-scalable-application/
98. Rubric for System Design Interviews - Exponent, accessed on May 1, 2025, https://www.tryexponent.com/courses/system-design-interviews/system-design-interview-rubric
99. The Architecture of Open Source Applications (Volume 2)Scalable Web Architecture and Distributed Systems, accessed on May 1, 2025, https://aosabook.org/en/v2/distsys.html
100. How to Build Scalable Websites and Apps - YouTube, accessed on May 1, 2025, https://m.youtube.com/watch?v=b449l8Vfu0A&t=0s
101. Website Rubric & Programming Rubric - Lloyd and Lauren, accessed on May 1, 2025, https://laurenandlloyd.com/lloyd-stuff/website-rubric-and-programming-rubric/
102. Guidance for Building a Containerized and Scalable Web Application on AWS, accessed on May 1, 2025, https://aws.amazon.com/solutions/guidance/building-a-containerized-and-scalable-web-application-on-aws/
103. Application Scalability: Ensuring Performance and Reliability - vFunction, accessed on May 1, 2025, https://vfunction.com/blog/application-scalability/
104. Decoding Key Challenges in Large-Scale Web App Development - Imaginovation, accessed on May 1, 2025, https://imaginovation.net/blog/large-scale-web-app-development-challenges-solutions/
105. Patterns for scalable and resilient apps | Cloud Architecture Center - Google Cloud, accessed on May 1, 2025, https://cloud.google.com/architecture/scalable-and-resilient-apps
106. Advanced Data Structures — Schedule, accessed on May 1, 2025, https://jeffe.cs.illinois.edu/teaching/datastructures/projects.html
107. data-analytics/Google Advanced Data Analytics Specialization/Foundations of Data Science/Assess your Course 1 end-of-course project.MD at main - GitHub,

accessed on May 1, 2025, https://github.com/alsobihi/data-analytics/blob/main/Google%20Advanced%20Data%20Analytics%20Specialization/Foundations%20of%20Data%20Science/Assess%20your%20Course%201%20end-of-course%20project.MD
108. Teaching Responsible Computing - C4SG@UB | - University at Buffalo, accessed on May 1, 2025, https://c4sg.cse.buffalo.edu/projects/Teaching%20Responsible%20Computing.html
109. 6.851: Advanced Data Structures (Spring'12) - courses - MIT, accessed on May 1, 2025, https://courses.csail.mit.edu/6.851/spring12/
110. Advanced Programming and Data Structure | La Salle | Campus Barcelona, accessed on May 1, 2025, https://www.salleurl.edu/en/programacion-avanzada-y-estructura-de-datos-1
111. Projects in Advanced Data Structures (6.851) - courses - MIT, accessed on May 1, 2025, https://courses.csail.mit.edu/6.851/spring21/project/
112. CSCI 151 - Data Structures - Hendrix College Computer Science, accessed on May 1, 2025, https://hendrix-cs.github.io/csci151/
113. Types of Rubrics | Rubrics | Feedback & Grading | Teaching Guides | Teaching Commons | DePaul University, Chicago, accessed on May 1, 2025, https://resources.depaul.edu/teaching-commons/teaching-guides/feedback-grading/rubrics/Pages/types-of-rubrics.aspx
114. Project Rubric - CIS 552, accessed on May 1, 2025, https://www.seas.upenn.edu/~cis5520/17fa/project-rubric.html
115. DM803: Advanced Data Structures, accessed on May 1, 2025, https://odinlister.sdu.dk/fagbesk/internkode/DM803/en
116. CIS 552, accessed on May 1, 2025, https://www.seas.upenn.edu/~cis5520/21fa/project-rubric.html
117. Rubric for Programming Assignments — CAPP 30122 - Computer Science with Applications II documentation, accessed on May 1, 2025, https://classes.cs.uchicago.edu/archive/2022/winter/30122-1/pa/rubric.html
118. Computer Science Assessment Rubric - ACM CCECC, accessed on May 1, 2025, http://ccecc.acm.org/guidance/computer-science/rubric/
119. CSCI 0014 - Data Structures - Search Results < Sierra College, accessed on May 1, 2025, https://catalog.sierracollege.edu/search/?P=CSCI%200014
120. Advanced Data Structures - UF CISE - University of Florida, accessed on May 1, 2025, https://www.cise.ufl.edu/~sahni/cop5536/grades.htm
121. Grading Rubric | PDF | Algorithms And Data Structures | Theoretical Computer Science, accessed on May 1, 2025, https://www.scribd.com/document/499491390/Grading-Rubric
122. CUYAMACA COLLEGE COURSE OUTLINE OF RECORD COMPUTER SCIENCE 282 – INTERMEDIATE JAVA PROGRAMMING AND FUNDAMENTAL DATA STRUCTURES, accessed on May 1, 2025, https://www.cuyamaca.edu/faculty-staff-resources/academic-senate/curriculum/outlines/files/cs/cs282.pdf
123. Top 100 Data Structure and Algorithms DSA Interview Questions Topic-wise,

accessed on May 1, 2025, https://www.geeksforgeeks.org/top-100-data-structure-and-algorithms-dsa-interview-questions-topic-wise/
124. Advanced Data Structures Explained - Amplitude, accessed on May 1, 2025, https://amplitude.com/explore/data/what-how-data-structure
125. Choosing the Right Data Structure: A Comprehensive Decision Guide - Design Gurus, accessed on May 1, 2025, https://www.designgurus.io/blog/choosing-the-right-data-structure-a-comprehensive-decision-guide
126. C# Most efficient data structure to insert and to remove lower half - Stack Overflow, accessed on May 1, 2025, https://stackoverflow.com/questions/36331129/c-sharp-most-efficient-data-structure-to-insert-and-to-remove-lower-half
127. What is Data Structure: Need, Types & Classification - Great Learning, accessed on May 1, 2025, https://www.mygreatlearning.com/blog/data-structure-tutorial-for-beginners/
128. CS 2413 – Data Structures – Spring 2025 - The University of Oklahoma, accessed on May 1, 2025, https://www.ou.edu/content/dam/coe/docs/cs/syllabi/2025-2026/CS_2413_Sridhar_Sp25.pdf
129. Organizing machine learning projects: project management guidelines., accessed on May 1, 2025, https://www.jeremyjordan.me/ml-projects-guide/
130. The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction, accessed on May 1, 2025, https://research.google/pubs/the-ml-test-score-a-rubric-for-ml-production-readiness-and-technical-debt-reduction/
131. 25 Machine Learning Projects for All Levels - DataCamp, accessed on May 1, 2025, https://www.datacamp.com/blog/machine-learning-projects-for-all-levels
132. LLM-Rubric: A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts† - arXiv, accessed on May 1, 2025, https://arxiv.org/html/2501.00274v1
133. 21 Machine Learning Projects [Beginner to Advanced Guide] - Greene Center, accessed on May 1, 2025, https://careereducation.rochester.edu/blog/2022/01/03/21-machine-learning-projects-beginner-to-advanced-guide/
134. APS360. Project Final Report Handout and Rubric, accessed on May 1, 2025, https://www.cs.toronto.edu/~lczhang/360/files/rubric_report.pdf
135. Let's Read Together - Testing Rubrics for Machine Learning - Chang Hsin Lee, accessed on May 1, 2025, https://changhsinlee.com/ml-tests/
136. The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction - Google Research, accessed on May 1, 2025, https://research.google.com/pubs/archive/aad9f93b86b7addfea4c419b9100c6cdd26cacea.pdf
137. What's the right machine learning approach to mark rubrics based on sequences of data?, accessed on May 1, 2025,

https://www.reddit.com/r/learnmachinelearning/comments/1drhs9x/whats_the_right_machine_learning_approach_to_mark/
138. Advanced Machine Learning Projects for data science - Train in Data's Blog, accessed on May 1, 2025, https://www.blog.trainindata.com/advanced-machine-learning-projects/
139. Rules of Machine Learning: | Google for Developers, accessed on May 1, 2025, https://developers.google.com/machine-learning/guides/rules-of-ml
140. Rubrics — AI for Education, accessed on May 1, 2025, https://www.aiforeducation.io/prompts/rubrics
141. Model evaluation, model selection, and algorithm selection in machine learning, accessed on May 1, 2025, https://sebastianraschka.com/blog/2016/model-evaluation-selection-part2.html
142. 【AML】Project Proposal Guidelines for Advanced Machine Learning Course - OpenReview, accessed on May 1, 2025, https://openreview.net/forum?id=0CfLQLw5yV
143. What's the right machine learning approach to mark rubrics based on sequences of data?, accessed on May 1, 2025, https://datascience.stackexchange.com/questions/129564/whats-the-right-machine-learning-approach-to-mark-rubrics-based-on-sequences-of
144. Evaluating data: How much training data do you need for machine learning?, accessed on May 1, 2025, https://kili-technology.com/training-data/how-much-data-do-you-need-for-machine-learning
145. Grading Rubric for AI Applications with Sergey Karayev (2019) - YouTube, accessed on May 1, 2025, https://www.youtube.com/watch?v=SO5IrU2Ff4Y
146. Advanced Machine Learning -- CSCI-GA.3033-052, accessed on May 1, 2025, https://cs.nyu.edu/~mohri/aml24/
147. Scoring rubric for application of ML concepts for image recognition -use stage, accessed on May 1, 2025, https://www.researchgate.net/figure/Scoring-rubric-for-application-of-ML-concepts-for-image-recognition-use-stage_tbl3_355087845
148. Classification in Machine Learning: A Guide for Beginners - DataCamp, accessed on May 1, 2025, https://www.datacamp.com/blog/classification-machine-learning
149. Lecture 4: Model Selection - ML Engineering, accessed on May 1, 2025, https://ml-course.github.io/master/notebooks/04%20-%20Model%20Selection.html
150. Advanced Machine Learning (PhD) Spring 2025, CMU 10716, accessed on May 1, 2025, https://www.cs.cmu.edu/~pradeepr/716/
151. MNIST preprocessing for the graded lab - Advanced Learning Algorithms - DeepLearning.AI, accessed on May 1, 2025, https://community.deeplearning.ai/t/mnist-preprocessing-for-the-graded-lab/347101
152. How to Choose a Machine Learning Model – Some Guidelines - DataScienceCentral.com, accessed on May 1, 2025,

https://www.datasciencecentral.com/how-to-choose-a-machine-learning-model-some-guidelines/
153. Suggestions for an intermediate level ML project : r/learnmachinelearning - Reddit, accessed on May 1, 2025, https://www.reddit.com/r/learnmachinelearning/comments/9elcte/suggestions_for_an_intermediate_level_ml_project/
154. Model Selection for Machine Learning - ScholarHat, accessed on May 1, 2025, https://www.scholarhat.com/tutorial/machinelearning/model-selection-for-machine-learning
155. How to Choose the Right Metric for Your Model - Codefinity, accessed on May 1, 2025, https://codefinity.com/blog/How-to-Choose-the-Right-Metric-for-Your-Model
156. Prerequisites and prework | Machine Learning - Google for Developers, accessed on May 1, 2025, https://developers.google.com/machine-learning/crash-course/prereqs-and-prework
157. CS2750: Machine Learning - University of Pittsburgh, accessed on May 1, 2025, https://people.cs.pitt.edu/~kovashka/cs2750_sp16/
158. www.google.com, accessed on May 1, 2025, https://www.google.com/search?q=popular+web+development+frameworks+2024
159. What are your 'must-have' tools in 2024 for efficient web development : r/webdev - Reddit, accessed on May 1, 2025, https://www.reddit.com/r/webdev/comments/1dgfodl/what_are_your_musthave_tools_in_2024_for/
160. Favorite Language and Framework or Stack for 2024 that is for "Speed of Development"? : r/webdev - Reddit, accessed on May 1, 2025, https://www.reddit.com/r/webdev/comments/1761jb6/favorite_language_and_framework_or_stack_for_2024/
161. 68 Most Popular Software Development Frameworks of 2024 - Code B, accessed on May 1, 2025, https://code-b.dev/blog/software-development-frameworks
162. List of Top Programming Languages and Frameworks for 2024 - Softermii, accessed on May 1, 2025, https://www.softermii.com/blog/top-programming-languages-and-frameworks-for-software-development
163. Top 15 Software Development Frameworks to Use in 2025 - Moon Technolabs, accessed on May 1, 2025, https://www.moontechnolabs.com/blog/software-development-frameworks/
164. Popular Software Engineering Frameworks Explained - Coding Blocks Blog, accessed on May 1, 2025, https://blog.codingblocks.com/2024/popular-software-engineering-frameworks/
165. Discussion of The 5 Must-Know Backend Frameworks in 2024 - DEV Community, accessed on May 1, 2025, https://dev.to/sufian/the-5-must-know-backend-frameworks-in-2024-3olg/com

ments
166. What's the most popular tech stack for Full Stack Developers in 2024? : r/webdev - Reddit, accessed on May 1, 2025, https://www.reddit.com/r/webdev/comments/1c1glpg/whats_the_most_popular_tech_stack_for_full_stack/
167. What is a rubric? A rubric is a scoring guide used to evaluate performance, a product, or a project. It has three parts: 1) per - Center For Teaching and Learning, accessed on May 1, 2025, https://ctl.utexas.edu/sites/default/files/build-rubric.pdf
168. Improve Model Accuracy with Data Pre-Processing - MachineLearningMastery.com, accessed on May 1, 2025, https://machinelearningmastery.com/improve-model-accuracy-with-data-pre-processing/
169. Items on grading rubric for a coding assignment - Computer Science Educators, accessed on May 1, 2025, https://cseducators.stackexchange.com/questions/6470/items-on-grading-rubric-for-a-coding-assignment
170. Grading Rubric : CS 252, accessed on May 1, 2025, https://cs.carleton.edu/faculty/jadrian/cs252/2014.0/rubric.html
171. iRubric: Algorithm Assignment rubric - Q24WAX2 - RCampus, accessed on May 1, 2025, https://www.rcampus.com/rubricshowc.cfm?code=Q24WAX2&sp=yes&
172. 2022 AP Exam Administration Scoring Guidelines - AP Computer Science Principles - AP Central, accessed on May 1, 2025, https://apcentral.collegeboard.org/media/pdf/ap22-sg-computer-science-principles.pdf
173. Traditional vs. Standards-based Grading - KnowledgeWorks, accessed on May 1, 2025, https://knowledgeworks.org/resources/traditional-grading-vs-standards-based-grading/