|  |  |
| --- | --- |
|  | |
| Book Name: | **Finite Abelian Groups, Elliptic Curves, Blockchain with Hashing and Graphs** |
| Manuscript Number: | **Ms\_BPR\_3842.5** |
| Title of the Manuscript: | **Introductory Essentials to Python Programming** |
| Type of the Article | **Book chapter** |

**General guidelines for the Peer Review process:**

This Book’s peer review policy states that **NO** manuscript should be rejected only on the basis of ‘**lack of Novelty’**, provided the manuscript is scientifically robust and technically sound.

To know the complete guidelines for the Peer Review process, reviewers are requested to visit this link:

<https://r1.reviewerhub.org/general-editorial-policy/>

**Important Policies Regarding Peer Review**

Peer review Comments Approval Policy: <https://r1.reviewerhub.org/peer-review-comments-approval-policy/>

Benefits for Reviewers: <https://r1.reviewerhub.org/book-benefits-for-reviewers>

|  |  |  |
| --- | --- | --- |
| PART 1: Review Comments | | |
| Compulsory REVISION comments | Reviewer’s comment | Author’s Feedback *(Please correct the manuscript and highlight that part in the manuscript. It is mandatory that authors should write his/her feedback here)* |
| **Please write a few sentences regarding the importance of this manuscript for the scientific community. Why do you like (or dislike) this manuscript? A minimum of 3-4 sentences may be required for this part.** | **This manuscript provides foundational insights into algorithmics and Python programming. It serves as an excellent educational resource for beginners and intermediates, introducing algorithm design, Python syntax, and practical programming constructs such as loops, functions, and data structures. Its clear explanation of fundamental concepts, coupled with illustrative examples and exercises, enhances its value for both teaching and learning.**  **I appreciate this manuscript for its structured approach and detailed coverage, which makes complex concepts accessible. However, the formatting and presentation could benefit from refinement to improve readability and engagement. Overall, it's a well-constructed and beneficial resource for the scientific and educational community.** |  |
| **Is the title of the article suitable?**  **(If not please suggest an alternative title)** | **The current title of the article, *"Introductory Essentials to Python Programming,"* is clear and informative but somewhat generic. If the manuscript also focuses on foundational algorithmic principles in addition to Python programming, the title could be enhanced to better reflect its comprehensive content.**  **Here are some alternative title suggestions:**   1. **"Foundations of Algorithmics and Python Programming"** 2. **"An Introductory Guide to Algorithms and Python Programming"** |  |
| Is the abstract of the article comprehensive? Do you suggest the addition (or deletion) of some points in this section? Please write your suggestions here. |  |  |
| **Are subsections and structure of the manuscript appropriate?** | **The subsections and structure of the manuscript are logically organized and provide a step-by-step introduction to both algorithmic principles and Python programming. However, there are areas where the structure can be refined for clarity and improved flow. Here's an analysis of its appropriateness:**  **Strengths:**   1. **Progressive Learning Curve: The manuscript begins with basic algorithmic concepts, transitions into Python programming, and concludes with exercises and solutions, creating a natural learning progression.** 2. **Subsections Are Well-Defined: Key topics like algorithmics, Python fundamentals, data structures, and exercises are broken into manageable sections, aiding readability.**   **Potential Improvements:**   1. **Relevance of Subsections:**    * **Some sections, such as *"Logigram,"* might not be immediately relevant to Python programming and could be moved to an appendix or integrated better with examples.**    * **Certain Python-related topics, like *Object-Oriented Programming (OOP),* could use more depth or practical examples to align with the rest of the content.** 2. **Integration of Exercises:**    * **Exercises and solutions are grouped at the end, which is helpful for reference but could be more effective if integrated within corresponding sections for better practice and understanding.** 3. **Flow and Connection:**    * **While the manuscript transitions well from algorithms to Python, some sections like *Arithmetic Operations and Operators* and *Manipulating Data Structures* could benefit from cross-references or a cohesive narrative linking them back to the overarching topic of algorithmics.**   **Suggested Structural Adjustments:**   1. **Consolidate Similar Topics:**    * **Merge sections like *Arithmetic Operations* and *Types of Basic Declarations* under a broader heading such as *Python Basics: Variables and Operations.*** 2. **Reorganize for Logical Flow:**    * **Place *Python Basics* immediately after the introduction to algorithms, followed by *Loops and Functions* and then *Data Structures.*** 3. **Add Contextual Introductions:**    * **Before diving into exercises, include a brief summary or transition discussing their importance in reinforcing the concepts covered.** 4. **Section Titles for Engagement:**    * **Use more engaging and descriptive titles, e.g., change *"Loops and Functions"* to *"Building Logic with Loops and Functions."***   **With these adjustments, the structure would align better with both pedagogical goals and the expectations of its target audience.** |  |
| **Please write a few sentences regarding the scientific correctness of this manuscript. Why do you think that this manuscript is scientifically robust and technically sound? A minimum of 3-4 sentences may be required for this part.** | **This manuscript demonstrates scientific correctness through its systematic and precise presentation of algorithmic concepts and Python programming fundamentals. The definitions of algorithms, algorithmics, and related constructs are consistent with widely accepted theoretical frameworks in computer science. Moreover, the inclusion of practical examples and Python code snippets reinforces the technical soundness of the content, as the examples align with Python's syntax and programming conventions.**  **The exercises and their solutions are also thoughtfully designed to test the understanding of key concepts, further validating the manuscript's rigor. Overall, the manuscript is scientifically robust, providing a solid foundation for learners and educators in programming and computational thinking** |  |
| **Are the references sufficient and recent? If you have suggestions of additional references, please mention them in the review form.**  **-** | **The manuscript's references are somewhat limited and may not fully encompass the latest advancements in Python programming and algorithmics. To enhance its comprehensiveness and relevance, consider incorporating the following recent resources:**   1. **"Data Structures and Algorithms with Python" by Kent D. Lee and Steve Hubbard (2024) This book offers an in-depth exploration of data structures and algorithms, complemented by Python code examples, serving as a valuable resource for both students and professionals.**   [**Springer Link**](https://link.springer.com/book/10.1007/978-3-031-42209-6?utm_source=chatgpt.com)   1. **"Machine Learning: Algorithms, Real-World Applications and Research Directions" by Sarker et al. (2021) This paper provides a comprehensive overview of machine learning algorithms and their practical applications, which could enrich discussions on algorithmic applications within the manuscript.**   [**Springer Link**](https://link.springer.com/article/10.1007/s42979-021-00592-x?utm_source=chatgpt.com)   1. **"An Empirical Analysis of Python Programming for Advanced Computing" (2022) This study examines Python's role in advanced computing, focusing on its applications in data science and machine learning, offering insights into Python's capabilities in these domains.**   [**IEEE Xplore**](https://ieeexplore.ieee.org/document/9823643?utm_source=chatgpt.com)   1. **"Python Algorithms: Mastering Basic Algorithms in the Python Language" by Magnus Lie Hetland (2020) This book delves into fundamental algorithms implemented in Python, providing practical examples and problem-solving techniques.**   [**Springer Link**](https://link.springer.com/book/10.1007/978-1-4842-0055-1?utm_source=chatgpt.com)   1. **"Functional Programming Paradigm of Python for Scientific Computation Pipeline Integration" (2024) This paper discusses the application of functional programming in Python for integrating scientific computation pipelines, highlighting modern programming paradigms.**   [**arXiv**](https://arxiv.org/abs/2405.16956?utm_source=chatgpt.com)  **Incorporating these references would provide readers with a broader perspective and deeper understanding of the current state of Python programming and algorithmics.** |  |
| Minor REVISION commentsIs the language/English quality of the article suitable for scholarly communications? | The language of the article is clear and comprehensible, but lacks the polish and precision expected in scholarly communications.  **Improvements in grammar, sentence structure, and technical terminology would enhance readability and professionalism.** |  |
| Optional/General comments |  |  |

|  |  |  |
| --- | --- | --- |
| **PART 2:** | | |
|  | **Reviewer’s comment** | **Author’s comment** *(if agreed with reviewer, correct the manuscript and highlight that part in the manuscript. It is mandatory that authors should write his/her feedback here)* |
| **Are there ethical issues in this manuscript?** | *(If yes, Kindly please write down the ethical issues here in details)* |  |

|  |  |
| --- | --- |
| **Reviewer Details:** | |
| Name: |  |
| Department, University & Country |  |