



ACE HACKER

Write Great Code

acehacker.com



ACE HACKER

Learn to write great code through immersive & hands-on classes. Sharpen your skills with programming competitions & hackathons. Connect & Collaborate with fellow coders, founders, and investors.

COURSES



Ace Hacker **Courses** are crafted to convey the most complex **STEM** concepts through engaging stories. Participate in these stories by engaging your thoughts, emotions, and imagination as you enter the narrative world and react to it. Tinker with the code in every story, push boundaries without fear of failure, build meaningful solutions to tackle real-world problems, and have fun along the way.



COMPETITIONS



Hack that code. Enhance your skills by participating in sprint **Programming Competitions** or marathon **Hackathons**. Beyond the intellectual growth, the thrill of competition adds an electrifying edge, giving you the chance to showcase your coding finesse and claim well-deserved bragging rights. The skills you hone and the memories you create will serve as lasting assets in your academic and professional pursuits.

CONFERENCES



Ace Hacker orchestrates **Colloquiums** and **Conferences** serving as a nexus for academicians, scholars, scientists, CXOs, and other luminaries in intellectual pursuits across diverse fields such as Artificial Intelligence, Quantum Computing, Blockchain, Cybersecurity, AR, VR, and Robotics, to name a few, with a mission to foster collaboration and knowledge exchange by bringing together the brightest minds to share their insights and research findings.





Mahabharatha

Through the Eyes of a Programmer

acehacker.com/learn/Mahabharatha



Master **Computer Programming Fundamentals**
and learn to write great code in **Python**
while exploring the epic tale of the
Mahabharatha!



Parva 1

Adi Parva: The Book of the Beginning

Covers the origins of the characters and the epic itself. Just as the Adi Parva introduces readers to the key characters, events, and themes that shape the saga, this orientation class familiarizes learners with fundamental concepts, syntax, and logical thinking needed for software development. The topics covered in these sessions set up a vast and complex environment filled with interconnected elements—characters and their relationships in the Mahabharatha, and data types, variables, and control structures in programming.

Introduction to Programming and Python Basics

- Overview of programming and problem-solving
- Python as a language for abstraction and computation
- Basics of Python syntax, variables, and data types
- Defining functions and exploring functional abstraction
- Exercises to write simple programs as expressions of computation





Parva 2

Sabha Parva: The Book of the Assembly Hall

Focuses on the grand assembly and the complexities of relationships, akin to understanding on how to build abstractions with functions. Just as the assembly hall is a place where diverse characters (each with unique roles) gather to deliberate, functions in programming bring together different operations and behaviors to work harmoniously. Each function, like each character, plays a specialized part, contributing to a larger program without needing to understand every detail about others. Through abstractions, programmers simplify complex tasks, much like the Sabha Parva highlights how individual motives and choices build up to create the epic's central conflicts and drive the story forward.

Building Abstractions with Functions

- Overview of programming and problem-solving
- Python as a language for abstraction and computation. Download Python.
- Basics of Python syntax, variables, and data types
- Defining functions and exploring functional abstraction
- Exercises to write simple programs as expressions of computation



Parva 3

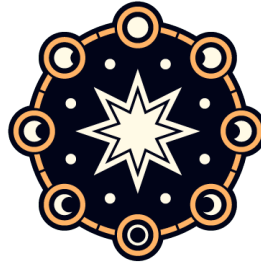
Vana Parva: The Book of the Forest

Depicts the Pandava's time in the forest. Just as recursion involves a function calling itself repeatedly with changing parameters until a base condition is met, the Pandavas continually face adversities, learning and growing through each encounter in the forest. Similarly, iterative processes loop through actions until a desired state is achieved, reflecting the repeated struggles and smaller victories the Pandavas experience as they move closer to the resolution of their exile. Both recursion and iteration allow programmers to break down complex problems into manageable cycles, mirroring the cyclical, layered challenges and personal transformations the Pandavas undergo in the forest.

Recursion and Iterative Processes

- Recursive functions and recursive problem-solving techniques
- Difference between recursive and iterative processes
- Tail recursion and its optimization
- Recursive vs iterative processes: practical and performance comparisons





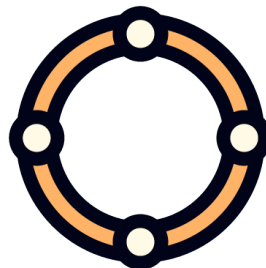
Parva 4

Virata Parva: The Book of Virata

Narrates the Pandavas' year spent in disguise at King Virata's court, where they assume different roles and cleverly hide their true identities. This part of the epic mirrors the programming concepts of Procedures and the Art of Decomposition. Just as the Pandavas break down their strengths into distinct roles—Arjuna as a dance instructor, Bhima as a cook, Yudhishtira as a royal advisor—programmers decompose complex problems into smaller, manageable procedures. Each procedure focuses on a specific task, contributing to the larger solution without revealing all details. In the same way, the Pandavas' individual disguises are specialized functions that help them collectively stay hidden and accomplish their shared goal. This chapter underscores how breaking down complexity into focused, modular parts can lead to a coherent, effective outcome, whether in code or in a strategic plan.

Procedures and the Art of Decomposition

- Decomposing complex problems into simpler functions
- Structuring programs with helper functions and subprocedures
- Procedural abstraction in large-scale code organization
- Python modules and namespaces for organized code



Parva 5

Udyoga Parva: The Book of Effort

Focuses on the preparation for war. The Udyoga Parva in the Mahabharatha, or "The Book of Effort", captures the intense diplomatic and strategic preparations that precede the great war, with alliances formed, resources gathered, and leaders choosing sides. This phase is akin to the programming concept of working with Compound Data, where complex data structures, like lists, dictionaries, and objects, allow different types of information to be organized, related, and managed together. Just as each kingdom brings its unique assets, motivations, and relationships into the impending conflict, compound data structures enable programmers to bundle varied yet interdependent data into cohesive units. Each alliance or diplomatic mission in Udyoga Parva reflects how diverse pieces of information—attributes, values, and connections—must interact seamlessly to shape a larger, strategic whole. Both in diplomacy and in code, organizing complex data effectively is essential to handle intricate interactions and prepare for major challenges ahead.

Working with Compound Data: Lists and Tuples

- Introduction to compound data types: lists and tuples
- Lists and their operations.
- Constructing and manipulating lists as sequences
- Understanding mutable vs immutable data structures
- Nested lists and multidimensional data structures





Parva 6

Bhishma Parva: The Book of Bhishma

Centers on the early days of the Kurukshetra war under the leadership of Bhishma, who commands the Kaurava forces with skill and restraint, concealing his own inner conflicts. This part of the epic aligns with the programming concepts of Data Abstraction and Information Hiding, where the internal workings of a system are hidden to reveal only what is necessary. Just as Bhishma keeps his personal loyalties and dilemmas hidden while presenting an unwavering front on the battlefield, data abstraction enables programmers to create clean interfaces, exposing only essential features to users while concealing complex underlying operations. This separation simplifies interactions with the system and allows it to function reliably, much like how Bhishma's leadership upholds the structure of the Kaurava forces while masking his true emotions. Both practices—information hiding and strategic concealment—maintain clarity and effectiveness within a larger, complex system.

Data Abstraction and Information Hiding

- Concept of data abstraction: separating interface from implementation
- Creating abstract data types (ADTs) in Python
- Constructors, selectors, and operations for data abstraction
- Encapsulation using functions and modules in Python



Parva 7

Drona Parva: The Book of Drona

Focuses on the battle strategies and intense combats under the command of Drona, a master of tactics and adaptability. This section parallels the programming concept of Higher-Order Functions in Depth, where functions can take other functions as arguments, return them as results, or manipulate them to create flexible, powerful operations. Drona's command style is similar, as he dynamically adapts strategies, leveraging the unique skills of different warriors, much like a higher-order function manipulates and coordinates other functions to achieve complex results. Just as higher-order functions abstract intricate tasks, simplifying control while enhancing functionality, Drona's sophisticated tactics allow him to orchestrate varied forces, adapting to each twist in battle. Both concepts emphasize flexibility, strategic delegation, and the power of abstraction to handle complexity effectively.

Higher-Order Functions in Depth

- Exploring the power of higher-order functions
- Using map, filter, and reduce for data transformations
- Function composition and currying techniques
- Applying higher-order functions to solve real-world problems





Parva 8

Karna Parva: The Book of Karna

Revolves around Karna's leadership on the battlefield, marked by his symbolic role as a warrior embodying duty, loyalty, and the quest for recognition. This parallels the programming concepts of Symbolic Expressions and Basic Expressions, where symbolic expressions represent abstract ideas or variables, and basic expressions perform core operations within a program. Karna, much like a symbolic expression, represents the ideals and conflicts of identity and fate, while his actions—powerful yet constrained by circumstance—are akin to basic expressions executing fundamental operations. Together, these expressions construct a complex narrative, just as symbolic and basic expressions combine to build more sophisticated code. Both Karna's role and programming expressions demonstrate how fundamental components and symbolic meanings interlace to create deeper, multi-layered structures and outcomes.

Symbolic Expressions and Basic Expressions

- Representing symbolic data in Python
- Working with strings and expressions as data
- Creating a simple symbolic calculator
- Evaluating symbolic expressions and handling errors
- Practical application: symbolic differentiation



Parva 9

Shalya Parva: The Book of Shalya

Describes the final phase of the Kurukshetra war, where shifting loyalties, exhaustion, and the fluctuating morale of the warriors profoundly impact the battle's outcome. This part of the epic mirrors the programming concepts of Modeling with State: Mutable Data and State, where variables and data can change over time, affecting the program's behavior dynamically. Just as the warriors' spirits and allegiances shift—particularly Shalya, who is reluctant in his role as Karna's charioteer—the mutable state in programming reflects how changing conditions and variables reshape processes. In both contexts, the concept of state introduces fluidity; in Shalya Parva, this fluidity influences each move on the battlefield, while in programming, mutable data allows systems to adapt and respond to evolving inputs, making the entire system more complex and responsive.

Modeling with State: Mutable Data and State

- Understanding mutable vs immutable objects
- Variables and assignment as representations of state
- Effects of mutable data on program behavior and design
- Simulating environments with stateful variables





Parva 10

Sauptika Parva: The Book of the Sleeping Warriors

Recounts the tragic night raid by Ashwatthama, who, driven by vengeance, slays the Pandava army's sleeping warriors. This destructive act is a powerful analogy for the programming concepts of Assignment and Environment Models, where values are assigned to variables within a specific environment, determining the scope and impact of each action. Just as Ashwatthama's choices operate within a particular environment of grief and revenge, assigning devastating consequences to his actions, assignments in programming directly affect the environment, binding variables to values that shape the program's state. Both illustrate how targeted actions within a defined context can lead to irreversible outcomes, highlighting the importance of controlled assignments and awareness of the broader environment in both programming and moral decisions.

Assignment and Environment Models

- Deep dive into the environment model of evaluation
- Frames, bindings, and scope in nested environments
- Local vs global variables and the nonlocal keyword
- Analyzing how Python manages function scope and variable binding



Parva 11

Stri Parva: The Book of the Women

The Stri Parva in the Mahabharatha, or "The Book of the Women", mournfully captures the aftermath of the war, highlighting the grief and resilience of the women who have lost loved ones. This part of the epic aligns with the programming concepts of Designing with Objects and Classes (Object-Oriented Programming), where each object represents an entity with its own attributes and behaviors, while classes define their shared characteristics. Each character—whether Gandhari, Draupadi, or Kunti—represents an instance of a "war survivor" class, embodying unique responses to loss but sharing the common traits of strength and sorrow. Just as OOP organizes complex systems by modeling distinct yet interconnected objects, Stri Parva brings together diverse perspectives, showing individual experiences within a collective tragedy. Both OOP and this part of the epic reveal how designing with distinct entities that interact and impact each other creates a nuanced, multifaceted narrative or system.

Designing with Objects and Classes (Object-Oriented Programming)

- Basics of object-oriented programming in Python
- Defining classes and creating instances
- Understanding attributes, methods, and encapsulation
- Polymorphism, inheritance, and method overriding





Parva 12

Shanti Parva: The Book of Peace

Focuses on the aftermath of the war, emphasizing the need for reconciliation, governance, and the establishment of a just society under Yudhishtira's rule. This transition reflects the programming concepts of Message Passing and Encapsulation, where objects communicate through well-defined messages while keeping their internal states hidden from the outside world. Just as the characters in Shanti Parva engage in dialogues and exchanges of wisdom to resolve conflicts and foster understanding, message passing allows different components of a program to interact without revealing their internal workings, promoting modularity and clarity. The encapsulation of complex issues into manageable dialogues mirrors how encapsulated data in programming simplifies interactions, allowing for smoother communication and problem-solving. Both contexts illustrate how constructive communication and clear boundaries facilitate peace and functionality, whether in governance or in the architecture of software systems.

Message Passing and Encapsulation

- Implementing object behavior through message passing
- Encapsulating state with private attributes
- Designing classes that interact through well-defined interfaces
- Comparing message passing to function-based design



Parva 13

Anushasana Parva: The Book of the Instructions

Features Bhishma imparting timeless wisdom to Yudhishtira, addressing questions of morality, duty, and governance in a gradual, responsive manner as Yudhishtira seeks guidance. This is akin to the programming concepts of Streams and Lazy Evaluation, where values are computed only as needed, allowing potentially infinite sequences to be processed efficiently and incrementally. Just as Bhishma's teachings flow continuously in response to Yudhishtira's queries—delivering wisdom in manageable, context-relevant portions—streams in programming generate data on demand, reducing memory usage and unnecessary computations. This controlled, lazy approach to evaluation ensures resources are used only when necessary, paralleling how Bhishma's incremental teachings offer focused insights, without overwhelming Yudhishtira, fostering deeper understanding over time.

Streams and Lazy Evaluation

- Introduction to streams as lazy lists
- Constructing infinite sequences using generators in Python
- Benefits and challenges of lazy evaluation
- Leveraging Python's generators (yield)
- Simulating lazy evaluation and functional data flows in Python





Parva 14

Ashvamedhika Parva: The Book of the Horse Sacrifice

Recounts Yudhishtira's Ashvamedha ritual, where a consecrated horse roams freely to assert sovereignty, leading to symbolic and diplomatic encounters as each region acknowledges Yudhishtira's rule. This is akin to Metaprogramming and Functional Abstractions in programming, where code dynamically generates or manipulates other code, creating abstract functions to adapt to various contexts. Just as the Ashvamedha horse acts as a vehicle for symbolic assertions of power across different kingdoms, metaprogramming enables code to adapt flexibly, allowing the programmer to write functions that apply across a wide range of scenarios. Functional abstractions let programs operate with minimal modifications, echoing how the Ashvamedha ritual consolidates authority while allowing for localized interpretations. Both metaprogramming and the Ashvamedha ritual illustrate how higher-level abstractions facilitate adaptability and influence across complex, dynamic systems.

Metaprogramming and Functional Abstractions

- Metaprogramming and its use in creating flexible abstractions
- Using decorators to add behavior to functions
- Exploring closures for encapsulating state within functions
- Functional composition and chaining operations



Parva 15

Ashramavasika Parva: The Book of the Hermitage

Reflects the latter years of Dhritarashtra, Gandhari, Kunti, and Vidura, who retreat into the forest seeking peace and introspection after the war. This phase of quiet reflection and the quest for understanding parallels the process of Building an Interpreter, where raw expressions are parsed, tokenized, and evaluated to derive meaning. Just as the elders in the story seek to parse the events of their lives, breaking down experiences to seek inner clarity and closure, an interpreter breaks down code into tokens, parses syntax, and evaluates expressions step-by-step. In programming, building an interpreter involves constructing a "language" that can understand and evaluate complex expressions, similar to how the elders interpret and seek wisdom from past events. Both processes require extracting meaning from raw input—whether it's life experiences or coded expressions—and transforming it into structured understanding through systematic interpretation.

Building an Interpreter

- Overview of interpreters and how they work
- Parsing simple arithmetic expressions in Python
- Tokenizing and syntax parsing using recursive descent
- Evaluating parsed expressions using a Python function
- Extending the interpreter to support variables and assignment
- Environment model for storing variable bindings
- Adding control structures (**if**, **while**) to the interpreter
- Scoping and managing environments within the interpreter
- Building a small language within Python for arithmetic expressions
- Developing a mini-interpreter capable of basic Python-like code execution





Parva 16

Mausala Parva: The Book of the Clubs

Tells the story of the Yadava dynasty's tragic self-destruction, spurred by internal conflicts and a prophecy of inevitable demise. This chaotic and unforeseeable end aligns with the concept of Nondeterministic Computing, where outcomes are not predictable from initial conditions and multiple possible paths may lead to different results. Just as the Yadavas' downfall unfolds in a way that defies precise prediction—rooted in random events and escalating conflicts—nondeterministic computing models scenarios where various computations can yield different solutions, depending on paths taken and external influences. In both, events unfold through a series of unpredictable choices and interactions, underscoring how randomness and nondeterminism can shape systems, be they human societies or computational models, in ways that are complex and often irreversible.

Nondeterministic Computing

- **Amb** and Search
- Examples of Nondeterministic Programs
- Nested Conditions
- Randomization and Probability using Python's `random` module
- Backtracking
- Implementing the **Amb** Evaluator



Parva 17

Mahaprasthanika Parva: The Book of the Great Journey

Narrates the final journey of the Pandavas (and a dog) toward the Himalayas, where each character gradually leaves the earthly plane, reaching their end in parallel but along shared paths. This reflects the programming concepts of Concurrency and Parallelism, where multiple tasks are processed in tandem, each following its own path but contributing to a unified goal. Just as the Pandavas, though traveling together, confront their individual fates at different moments, concurrent tasks in programming can execute independently yet share resources or synchronize at certain points to achieve a common purpose. Both the journey and parallel computing illustrate how individual processes can operate simultaneously, each reaching completion in its own time while contributing to the broader progression of the whole.

Concurrency and Parallelism

- Basics of concurrency and parallelism concepts
- Working with threads and processes in Python (threading and multiprocessing)
- Understanding race conditions and deadlock
- Using `asyncio` for asynchronous programming





Parva 18

Svargarohana Parva: The Book of the Ascent to Heaven

Depicts Yudhishtira's final transcendence to the heavens, symbolizing the culmination of his earthly journey and the elevation of his essence to a higher, purified state. This is akin to the programming concept of Compilation—the process of transforming human-readable code into a more efficient, machine-understandable format, ready for execution. Just as Yudhishtira's life experiences are refined and elevated, the compilation process converts high-level code into optimized, low-level instructions through parsing, optimization, and code generation. Compilation transforms complex, symbolic instructions into executable machine language, mirroring how Yudhishtira's earthly knowledge is distilled and perfected, preparing him for his final, transcendent destination. Both journeys—from source code to binary, and from earthly realm to heaven—represent transitions that yield a complete, efficient end state.

Compilation and the inner workings of a Compiler

- Structure of the Compiler
- Compiling Expressions
- Compiling Combinations
- Compiling Instruction Sequences
- Lexical Addressing
- Interfacing compiled code to the evaluator

Write Great Code

Choose a schedule that works best for you. Join from anywhere across the globe. Ace Hacker is a Hybrid / Hyflex Classroom learning environment. The classroom includes a mix of students who are present onsite and those who wish to join the class virtually.

Weekdays



Mon, Tue, Wed, Thu, Fri
9 Weeks
1.5 hours/day

Weekends



Saturdays & Sundays
9 Weeks
4 hours/day

Custom Plan



Have a team of 9 or more members and need a custom training plan?
Email:
connect@acehacker.com
Call: (+91) 988.011.2117





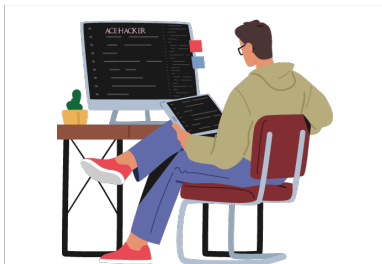
Mahabharatha

Through the Eyes of a Programmer

acehacker.com/learn/Mahabharatha

SHARPEN YOUR AXE

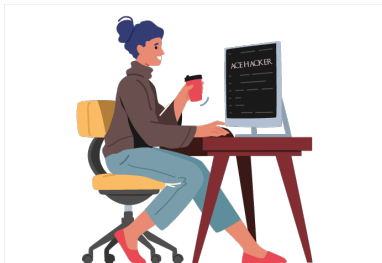
Use Lab Exercises, Projects, Coding Competitions, and Hackathons as opportunities to practice and apply your programming skills in real-world scenarios.



To augment your proficiency in Programming, you will be regularly challenged with various assignments like coding challenges, algorithmic puzzles, trick logical, analytical, and mathematical problems - designed to enable you to think creatively and outside the box.

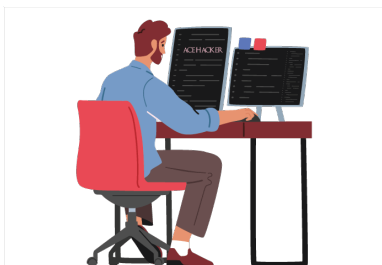
Expect 4 levels of difficulty in these assignments:

- **Beginner:** Exercises that test your understanding of the subject.
- **Intermediate:** Exercises that add new and thought-provoking information to the subject.
- **Advanced:** Exercises that are intended to challenge you.
- **Expert:** Exercises that are extremely difficult by comparison with most others.



The projects are designed to ensure that you not only understand the theoretical concepts of Computer Programming also gain hands-on experience in applying those concepts to real-world scenarios. You'll be required to collaborate as a team to attempt some projects while working as a Lone Wolf / individual contributor on others.

- **Four Cornerstone projects** that will reinforce in you the strong foundational knowledge of Cybersecurity & Cyber Warfare.
- **Two Keystone projects** will be more challenging, building on the knowledge you have gained through the Cornerstone projects.
- **One Capstone project**, which will be the culmination of your learning experience in this course.



Competitions and Hackathons are a great way to reinforce your learning and to challenge you to apply your skills to real-world scenarios. By participating in these events, you will gain practical experience and develop your problem-solving skills. You will be participating in a variety of events, including but not limited to:

- **Sprint Coding Competitions**
- **Marathon Hackathons**
- **Catch-the-Flag (CTF) challenges**
- **War Games**

In some competitions, you will collaborate as a team, which will aid you in developing your teamwork skills. In other competitions, you will participate as a Lone Wolf, which will challenge you to think independently and to rely on your own skills and knowledge.





Mahabharatha

Through the Eyes of a Programmer

acehacker.com/learn/Mahabharatha

CERTIFICATE IN PROGRAMMING

Based on your performance you either get a **Certificate of Excellence** or **Certificate of Completion** after successful completion of the course.



CERTIFICATE OF EXCELLENCE



CERTIFICATE OF COMPLETION



ACE THAT PROGRAMMING INTERVIEW

As a part of this course, you'll learn to crack Programming interviews. You'll be thoroughly trained using:

- **Mock Interviews:** We'll simulate the pressure and format of a real coding interview, allowing you to practice and improve your technical, communication, and presentation skills under similar conditions.
- Through **Whiteboarding**, you'll learn to visually represent your thought process on a physical or a digital whiteboard.
- You'll learn how to handle **Impossible Questions & Kobayashi Maru situations** which are essential in a coding interview to show adaptability, creativity, and resilience under pressure.
- Learn to solve **Algorithmic, and Mathematical and Puzzles** using Python, C, C++, Go, ARM and develop a creative & analytical mindset, logical reasoning, and problem-solving skills that are crucial in cracking a Programming interview.





Mahabharatha

Through the Eyes of a Programmer

acehacker.com/learn/Mahabharatha



Need more information?
Contact us.

- URL: <https://acehacker.com/learn/Mahabharatha>
- connect@acehacker.com
- (+91) 988.011.2117

