Perl Data Structures - 18-Week Teaching Plan

Week-by-Week Breakdown:

1. **Introduction to Perl & Data Structures**
   - Perl syntax overview (comparison with C)
   - Scalars, arrays, hashes, and references
   - Articles:
     - [Perl Basics](https://perldoc.perl.org/perlintro)
     - [Perl References and Data Structures](https://perldoc.perl.org/perlreftut)

2. **Arrays and List Processing**
   - Array operations: push, pop, shift, unshift
   - Slices, sorting, and transformations
   - Article: [Perl Arrays](https://perldoc.perl.org/perldata)

3. **Hashes: Perl's Built-in Hash Table**
   - Hash operations, usage, and best practices
   - Example: Implementing a frequency counter
   - Article: [Perl Hashes](https://perldoc.perl.org/perlfunc#Hash-Functions)

4. **References and Nested Data Structures**
   - Pointers vs. references (comparison with C)
   - Multi-dimensional arrays and hashes of hashes
   - Article: [Advanced Data Structures in Perl](https://perldoc.perl.org/perldsc)

5. **Stacks (LIFO)**
   - Implementing stacks using arrays
   - Using packages and subroutines for modular design
   - Article: [Perl Stack Implementation](https://www.perlmonks.org/?node_id=900900)

6. **Queues (FIFO)**
   - Implementing queues with arrays
   - Circular queues and performance considerations
   - Article: [Perl Queue Implementation](https://www.perlmonks.org/?node_id=1028569)

7. **Linked Lists**
   - Implementing singly and doubly linked lists in Perl
   - Comparison with C's pointer-based implementation
   - Article: [Perl Linked Lists](https://www.perlmonks.org/?node_id=286445)

8. **Trees and Binary Search Trees (BST)**
   - Tree structures using hashes and references
   - Implementing insert, delete, and traversal
   - Article: [Perl Binary Trees](https://www.perlmonks.org/?node_id=166097)

9. **Heap and Priority Queue**
   - Implementing heaps using arrays
   - Using CPAN's `Heap::Simple`
   - Article: [Heap in Perl](https://metacpan.org/pod/Heap::Simple)

10. **Graphs and Graph Traversals**
    - Representing graphs using adjacency lists
    - BFS and DFS implementation
    - Article: [Graph Algorithms in Perl](https://www.perlmonks.org/?node_id=379374)

11. **Sorting Algorithms**
    - Implementing Bubble, Merge, and QuickSort in Perl
    - Benchmarking and optimization
    - Article: [Sorting in

Perl](https://www.perlmonks.org/?node_id=236206)

12. **Hashing and Bloom Filters**
    - Custom hash functions
    - Implementing Bloom filters for efficient lookups
    - Article: [Perl Bloom Filters](https://www.perlmonks.org/?node_id=11104506)

13. **Tries and String Matching Algorithms**
    - Building a trie for dictionary lookup
    - Implementing Knuth-Morris-Pratt (KMP) and Rabin-Karp
    - Article: [Perl Trie Implementation](https://www.perlmonks.org/?node_id=627977)

14. **Perl's Built-in Data Handling (DBM, Storable, JSON)**
    - Using `DB_File` and `Storable` for persistent storage
    - Working with JSON data structures
    - Article: [Storable Module](https://perldoc.perl.org/Storable)

15. **Object-Oriented Perl for Data Structures**
    - Implementing data structures as Perl objects
    - Using `bless` and encapsulation
    - Article: [Object-Oriented Perl](https://perldoc.perl.org/perlobj)

16. **Perl Modules and CPAN for Data Structures**
    - Exploring CPAN modules for common data structures
    - Installing and using `Tie::Hash` for custom hash behaviors
    - Article: [Perl CPAN Modules for Data Structures](https://www.perlmonks.org/?node_id=27653)

17. **Final Project: Implementing a Custom Data Structure**
    - Assign students a unique problem to solve
    - Example: Implementing an LRU Cache or a Social Network Graph
    - Article: [LRU Cache in

Perl](https://www.perlmonks.org/?node_id=1172807)

18. **Review, Optimization, and Future Learning Paths**
    - Profiling and debugging Perl data structures
    - Advanced concepts: Memoization, Lazy Evaluation, Functional Perl
    - Articles:
      - [Profiling Perl Code](https://www.perlmonks.org/?node_id=235766)
      - [Functional Perl](https://www.perl.com/article/functional-programming-in-perl/)