**Section 1: Go Programming languages**

1. <https://www.geeksforgeeks.org/golang/?ref=lbp>
2. Deep dive data structures in GoLang **(Arrays,** **Slices, Maps, Structs, Interfaces ) – Practice programs**
3. Integrate pProfs for programs you write - <https://jvns.ca/blog/2017/09/24/profiling-go-with-pprof/>
4. <https://www.golangprograms.com/basic-programs.html> **( Execute every single program here)**
5. <https://golangbyexample.com/golang-comprehensive-tutorial/>
6. <https://golongwithgolang.com/when-should-you-use-pointers-in-go>
7. <https://golongwithgolang.com/thread-safety-in-golang>
8. <https://golongwithgolang.com/the-glorious-nethttp-go-module-1>
9. <https://www.geeksforgeeks.org/how-to-use-go-with-mysql/>

**Section 2: Go Libraries**

1. GoRoutines in Golang
   1. <https://www.golangprograms.com/goroutines.html>
   2. https://www.digitalocean.com/community/tutorials/how-to-run-multiple-functions-concurrently-in-go
   3. <https://www.golangprograms.com/go-language/channels.html>
   4. <https://www.golangprograms.com/go-language/concurrency.html>
   5. <https://blog.logrocket.com/concurrency-patterns-golang-waitgroups-goroutines/#:~:text=Golang%20provides%20goroutines%20to%20support,of%20stack%20space%20to%20initialize>.
2. GORM in Golang
   1. <https://gorm.io/docs/index.html>
   2. <https://www.mindbowser.com/golang-go-with-gorm/>
   3. <https://towardsdev.com/introduction-to-orm-using-gorm-in-golang-d1936a45ffdb>
   4. <https://golangdocs.com/gorm-golang-orm-package>
   5. <https://www.youtube.com/watch?v=dpx6hpr-wE8>
3. Go Gin
   1. <https://semaphoreci.com/community/tutorials/building-go-web-applications-and-microservices-using-gin>
   2. <https://blog.logrocket.com/building-microservices-go-gin/>
   3. <https://blog.logrocket.com/how-to-build-a-rest-api-with-golang-using-gin-and-gorm/>
4. Golang Mux Router
   1. <https://gowebexamples.com/routes-using-gorilla-mux/>
   2. <https://golangdocs.com/golang-mux-router>
   3. <https://medium.com/geekculture/develop-rest-apis-in-go-using-gorilla-mux-5869b2179a18>
   4. <https://semaphoreci.com/community/tutorials/building-and-testing-a-rest-api-in-go-with-gorilla-mux-and-postgresql>
   5. <https://www.golangprograms.com/goroutines-and-channels-example.html>

**Section 3: Caching and Data streaming tools**

1. Caching
   1. <https://www.geeksforgeeks.org/what-is-the-caching-mechanism/>
   2. <https://medium.datadriveninvestor.com/all-things-caching-use-cases-benefits-strategies-choosing-a-caching-technology-exploring-fa6c1f2e93aa>
2. GoLang and caching mechanism
   1. <https://kislayverma.com/software-architecture/architecture-patterns-caching-part-1/>
   2. <https://www.mailgun.com/blog/it-and-engineering/golangs-superior-cache-solution-memcached-redis/>
   3. <https://github.com/mailgun/groupcache>
   4. <https://stackoverflow.com/questions/31389484/how-does-groupcache-in-go-compare-to-redis-and-memcached>
   5. <https://medium.com/@shreyanshsinha/thundering-herd-problem-cache-stampede-187a7b2f6cf7>
3. **Redis ( Pre requisite: Well versed with Data structures)** 
   1. <https://www.youtube.com/watch?v=Qu5gX2uOaL8>
   2. <https://tutorialedge.net/golang/go-redis-tutorial/>
   3. <https://github.com/gomodule/redigo>
4. KAFKA
   1. <https://www.tutorialspoint.com/apache_kafka/apache_kafka_introduction.htm>
   2. <https://www.sohamkamani.com/golang/working-with-kafka/>
   3. <https://github.com/confluentinc/confluent-kafka-go>

Videos:

1. <https://www.youtube.com/watch?v=jFfo23yIWac>
2. <https://youtu.be/YS4e4q9oBaU>

Assignment – Create a Micro Finance Service

The Micro Finance Service should run on Go and should support the following functionality

1. Authentication (Can use any Authentication Method)
2. Create Account (Name/Location/PAN/Address/Contact Number/Sex/Nationality)
3. Update Account
4. Delete Account
5. Login/Logout
6. Transactions (Credit / Debit)

For the DB, you can maintain a connection pool (say max 10) and use the pool for all the transactions.

Tech to be used:

Redis to be used for Authentication Token (Generation/Validation)

DB – Any SQL DB

API Backend – Go Lang

Test Cases:

1. Write Unit Test cases for verifying Create/Delete Accounts
2. Write Unit Test cases for Verifying Login/Logout
3. Write Unit Test cases for Credit/Debit Transactions
4. Write Test Cases to Verify Connection Pool of DB CRUD operations.

**Day 4:**

1. Binary Search: Searching an element using binary search method.
2. Count Zeros: Count the number of zeros in an array.
3. Floor of a number: Largest element in an array smaller than or equal to x.
4. Index of extra element: Finding the index of the missing element which is present in the main array.
5. Kth smallest element: finding the Kth smallest element in a given array.
6. Majority element: Element appearing max number of times.
7. Missing Number: Guess the missing number in an array.
8. Peak element: Element with highest value.
9. Search element: Searching a given element in an array
10. Search element in sorted array: Searching for a given element in a sorted array.

**Day 5:**

1. Second highest element: Returning the second highest element in a given array.
2. Square root of a given number: Finding the square-root of a given number.
3. Sub-array sum: Finding a subarray whose sum is equal to a given value.
4. Sub-b array: Check if a given array is a subset of the main array.
5. Transition: Index at which change of values happens in an array.
6. Triplets sum zero: Find the triplets whose sum is equal to zero
7. Value equal to index: Find the element whose value is equal to its index.
8. Alternate elements : Return only the alternate elements in an array.
9. Intersection of arrays : Return the common elements between two arrays.
10. Array leader: Return the highest element of a given array.
11. Array rotate: Rotate an array with a specified number of elements.
12. Array wave: Return a sorted array in the form of a wave i.e one small element and the next is greater.
13. Count pairs: Return pairs with the given sum.
14. Equilibrium point : The index where the left and right side elements add up to the same sum.
15. Kadane's Algorithm : Using Kadanes's algorithm to find the maximum sum of elements of a sub array.
16. Min and Max: Finding the smallest and largest element in a given array
17. Minimum distance: The distance between two given elements in an unsorted array.
18. Palindrome element: Find if a given element in an array is palindrome or not
19. Search element: Return the index of a given element.
20. Union of two arrays: Return the non-repetitive elements of two arrays.

**Day 6**

1. Check sorted: Check if a given array is sorted or not.
2. Max occurring element: Return the element that is occurring the maximum number of times.
3. Median of two arrays: Merge and sort two arrays and return the middle element.
4. Move zeros to end: Return an array with all the zeros moved to the right side.
5. Negative values to left: Return an array with all the negative values moved to the left.
6. Number divisible by 3:Return the element which is divisible by 3 in a given array.
7. Pythagorean Triplet: Return 1 if the given triplet forms a pythagorean triplet.
8. Smallest subarray with sum: Return the smallest subarray with the given sum.
9. Anagram: Check if both the strings have the same elements.
10. First non-repeating character: Return the first non-repeating character of a given string.
11. IP address validation: Check if the given IP address is valid or not.
12. Longest palindrome substring: Print the longest palindromic substring in a given string.
13. Multiply two strings: Multiplying two given strings.
14. Permutation of string: Returning the value of permutation of a string.
15. Reverse without reversing each word: Reverse just the words of a sentence.
16. Roman to integer: Convert Roman numbers to integers.
17. Rotate a string: Rotating a string by the given number of characters.
18. Check if string present in a string: Return true if the given string contains a set of characters.
19. Count duplicates: Count the number of duplicate characters in a string.
20. First letter :Return the first letter of each word.

**Day 7**

1. Isogram: Return true if the string has no repeating characters.
2. Max occurring : Return the character occurring maximum number of times in a string.
3. Remove duplicates: Remove the duplicate values in a string.
4. Run length decode: Print the Character the number of times the integer before it.
5. Run length encode: Print the number of times a character if found as integer before the character.
6. Sum of string: Print the sum of all characters of a string.
7. Extract numbers: Return the integers present in a string
8. First repeating character: Return the first duplicate character in a string.
9. Product even odd: Print if the product of two numbers is even or odd.
10. Remove special characters: Remove any special character in a string and print the remaining.
11. Reverse each word: Reverse each letter of all the words in a sentence
12. Sort string: Sort a string in alphabetical order.
13. Sort without function: Sort a string without using inbuilt function.
14. Subsequence: Return true if a string is a subsequence of another string.
15. Sum of two strings: Return the arithmetic sum of two strings.
16. Count alphabet: Return the number of alphabets in a given string.
17. Count last word: Return the count of characters in the last word of a string.
18. Longest word in dictionary: Return the longest word in a given dictionary.
19. Nearest multiple of 10: Return the nearest multiple of 10 of a given number.
20. Next even number: Return the next even number of a given number.

**Day 7**

1. Pangram: Return true if the sentence contains all the 26 alphabet.
2. Remove vowels: Remove all the vowels from a given string.
3. Word with max frequency: Return the word occurring maximum number of times.
4. Three consecutive strings: Remove a character if it is repeated 3 times consecutively.
5. Bubble sort: Sorting an array using bubble sort technique.
6. Combine the strings : Concatenate two strings
7. Merging unsorted : Merging two unsorted strings after sorting them.
8. Number of characters in a word : Return the number of characters in a string.
9. Number with d digit : In a list of numbers from 0 to n print the numbers with 'd' digit in it.
10. Sort 0,1,2: Sort the digits in ascending order
11. Even greater than odd : Return the largest numbers in even index and smallest numbers in odd index
12. Kth smallest element : Return the Kth smallest element in a given array.
13. nCr: Combination of given n and r.
14. Nth Fibonacci : Return the Nth digit of a fibonacci series.
15. Nth term of GP : Return the Nth term of a geometric progression.
16. Power of numbers: Return the power of reverse of a number raised to the number.
17. Replace 0 with 5: Return string with 0 replaced with 5.
18. Sort all except one: Return sorted list except the given index value.
19. Sort half sorted: Return a sorted array which is half sorted.
20. Sum of series: Return the sum of a series of numbers.

**Day 8**

1. Binary addition: Return the binary sum of given binary numbers.
2. Closest number: Return the closest number divisible by a given number.
3. LCM and GCD: Return the LCM and GCD of given numbers.
4. Multiplication table: Print the table of a given number.
5. Nth root of M: Return the Nth root of a given number.
6. Nth term of AP: Return the Nth term of given arithmetic progression.
7. Odd even: Print if the given number is odd or even.
8. Right most non zero element: Print the rightmost non zero number in the product of given numbers.
9. Sum of digit palindrome: Check if sum of all digits an array form a palindrome.
10. Swap 2 numbers:Swap the values of two variables without using the 3rd variable.
11. Alternate elements: Return only the alternate elements in an array.
12. Intersection of arrays: Return the common elements between two arrays.
13. Array leader :Return the highest element of a given array.
14. Array rotate: Rotate an array with a specified number of elements.
15. Array wave: Return a sorted array in the form of a wave i.e one small element and the next is greater.
16. Count pairs: Return pairs with the given sum.
17. Equilibrium point: The index where the left and right side elements add up to the same sum.
18. Queue (enqueue, dequeue, peek, isEmpty, clear, count, copy) operations.
19. Linked list ( insert, delete, recursive display, reverse using iteration, delete duplicates)
    1. Circular linked list (insert, delete) operations.
    2. Doubly linked list (insert, delete) operations.