Detailed Procedure:

--->(1st June to 15th June):

*Team interaction

*Report preparation

*C++STL/Java Collections

--->(15th June to 30th June):

*Introduction to Data structures

*Basic Number Theory(GCD etc.) and Maths required in Competitive programming

*Bitwise operators

*Arrays

*Strings

* Stack and Queues Implementation

--->(1st July to 10th July)

*Applications of stack and queues and problems based on them

*Linked Lists (Singly and doubly linked lists, circular linked list) implementation and problems based on them

*Time and Space complexity analysis

--->(11st July to 20th July)

*Searching algos (Linear and binary search)

* Sorting algos (Bubble, Merge, Quick,Selection,Heap sort etc.)

*Intro to recursion and divide and conquer approach

--->(21st July to 31th July)

*Hashing

*Greedy approaches

--->(1st August to 10th August)

*Dynamic programming

(11st August to 15th August)

*Priority Queues and heaps

*Backtracking

(15th August to 30th August)
*Trees and Graphs
*Wrapping up the course

(1st June to 15th June):

*Team interaction
*C++STL/Java Collections

Resources:Just go through these blogs once. No need to cram all the functions for now. You will get familiar with these when we will do questions in upcoming sessions. But read these once here as well.

C++ STL: https://www.hackerearth.com/practice/notes/standard-template-library/
Collections in Java: https://www.javatpoint.com/collections-in-java
For Python: https://www.tutorialspoint.com/python-tricks-for-competitive-coding

(15th June to 30th June):

*Introduction to Data structures
*Basic Number Theory(GCD etc.) and Maths required in Competitive programming
*Bitwise operators
*Arrays
*Strings
* Stack and Queues Implemention

Resources:
*Intro to DS:
https://www.geeksforgeeks.org/introduction-to-data-structures-10-most-commonly-used-data-structures/
https://medium.com/swlh/introduction-to-data-structures-9134b7d064a6
*Maths:
https://www.hackerearth.com/practice/math/number-theory/basic-number-theory-1/tutorial/
*Bitwise operators:
C++: https://www.geeksforgeeks.org/bitwise-operators-in-c-cpp/
Java: https://www.geeksforgeeks.org/bitwise-operators-in-java/

Python: https://www.geeksforgeeks.org/python-bitwise-operators/

For all: https://www.geeksforgeeks.org/bitwise-hacks-for-competitive-programming/

*Arrays:

https://www.geeksforgeeks.org/introduction-to-arrays/

C++: https://www.geeksforgeeks.org/arrays-in-c-cpp/

Java: https://www.geeksforgeeks.org/arrays-in-java/

Python: https://www.geeksforgeeks.org/array-python-set-1-introduction-functions/

https://www.geeksforgeeks.org/array-in-python-set-2-important-functions/?ref=lbp

*Stacks and Queues:

https://www.youtube.com/watch?v=BrVZZZkkGGI

https://www.youtube.com/watch?v=gnYM_G1ILm0

https://www.geeksforgeeks.org/stack-data-structure-introduction-program/

https://www.geeksforgeeks.org/queue-set-1introduction-and-array-implementation/

Solved problems: These are some of the important questions based on the topics of this session.First read the problem statement carefully and try to implement it on your own. Whenever you get stuck, take hints from the solution.So start now. Best of luck!

1.Pascal Triangle (Google it)

2.https://www.geeksforgeeks.org/swap-two-numbers-without-using-temporary-variable/

3.https://www.geeksforgeeks.org/a-product-array-puzzle/

4. https://www.geeksforgeeks.org/array-rotation/

5. https://www.geeksforgeeks.org/write-a-program-to-reverse-an-array-or-string/

6.https://www.geeksforgeeks.org/count-uppercase-lowercase-special-character-numeric-values/

7.Kadane's Algorithm

8.Count trailing zeros in factorial of a number

9.Check whether string is palindrome or not.

10. Stack and Queue implementation using arrays.

Unsolved problems:

1.https://www.hackerearth.com/practice/algorithms/string-algorithm/basics-of-string-manipulation/practice-problems/algorithm/remove-duplicates-3/

2. https://www.codechef.com/problems/ZOZ

3. https://www.codechef.com/problems/EXPCODE1

Some more problems and resources:

1. Check for balanced parentheses in an expression

2. https://www.hackerrank.com/challenges/maximum-element/problem
3. https://www.geeksforgeeks.org/implement-two-stacks-in-an-array/
4. https://www.geeksforgeeks.org/the-stock-span-problem/
5. Check whether two given strings are anagrams or not
6. https://www.codechef.com/LRNDSA02/problems/PSHOT
7. https://www.hackerrank.com/challenges/repeated-string/problem
8. https://www.hackerrank.com/challenges/time-conversion/problem
9. https://www.hackerrank.com/challenges/kangaroo/problem
10. https://www.hackerrank.com/challenges/divisible-sum-pairs/problem
11. https://www.hackerrank.com/challenges/designer-pdf-viewer/problem
12. Sieve of Eratosthenes(https://www.youtube.com/watch?v=1K-NPoepIHc)
13. https://practice.geeksforgeeks.org/problems/odd-divisors/0

(1st July to 10th July):

*Applications of stack and queues and problems based on them

*Linked Lists (Singly and doubly linked lists, circular linked list) implementation and problems based on them

*Time and Space complexity analysis

Resources:

Note: Refer to video tutorials or search other websites on google if you get stuck anywhere.

Linked List:

Study below topics from this site and also try to implement by hand on your own after reading these tutorials. : https://www.geeksforgeeks.org/data-structures/linked-list/

Singly Linked List: 1,2,3,4,5

Circular Linked List: 1,2,5,8,11

Doubly Linked List:1,2,12,26,27

Time and Space complexity analysis:

Playlist: https://www.youtube.com/playlist?list=PLEbnTDJUr_IeHYw_sfBOJ6gk5pie0yP-0

Watch lectures 1,2,3,4,5 from this playlist.

Notes: https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/#:~:text=While%20analyzing%20an%20algorithm%2C%20we,the%20length%20of%20the%20input.&text=We%20will%20only%20consider%20the%20execution%20time%20of%20an%20algorithm.

Practice problems:

Time and Space complexity analysis:

https://www.geeksforgeeks.org/practice-questions-time-complexity-analysis/

Linked List: Do all questions from below link (All are good questions)

https://www.hackerrank.com/domains/data-structures?filters%5Bsubdomains%5D%5B%5D=linked-lists

(11st July to 20th July):

*Searching algos (Linear and binary search)

* Sorting algos (Bubble, Merge, Quick,Selection,Heap sort etc.)

*Intro to recursion and divide and conquer approach

Sorting algos:

Resources:

Read below mentioned topics and implement(must-do) from this site:

https://www.geeksforgeeks.org/sorting-algorithms/

Topics: Selection sort,Bubble sort, Insertion sort,Merge sort,Quick sort,Counting sort

Other resources:

https://www.geeksforgeeks.org/time-complexities-of-all-sorting-algorithms/

https://www.geeksforgeeks.org/sort-c-stl/

https://www.geeksforgeeks.org/arrays-sort-in-java-with-examples/

https://www.geeksforgeeks.org/sort-in-python/

Coding Problems: Do below mentioned questions from this site.

https://www.geeksforgeeks.org/sorting-algorithms/

To-do: 1,4,14,52,85(Hard but try)

Quiz:https://www.geeksforgeeks.org/algorithms-gq/searching-and-sorting-gq/

Searching algos:

Resources:

https://www.geeksforgeeks.org/linear-search/

https://www.geeksforgeeks.org/binary-search/

https://www.geeksforgeeks.org/linear-search-vs-binary-search/

Practice problems:

Note: If a question has more than one approach to solve, then go through all of the approaches.

Coding problems:

1. **https://www.geeksforgeeks.org/find-the-missing-number/**
2. **https://www.geeksforgeeks.org/search-an-element-in-a-sorted-and-pivoted-array/**
3. **https://practice.geeksforgeeks.org/problems/square-root/1**
4. **https://practice.geeksforgeeks.org/problems/peak-element/1**
5. **Aggressive cows problem**
6. **https://www.hackerrank.com/challenges/sherlock-and-array**

Quiz:https://www.geeksforgeeks.org/algorithms-gq/searching-gq/

Recursion:

Resources:https://www.geeksforgeeks.org/recursion/

Coding problems:

https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/

https://practice.geeksforgeeks.org/tag-page.php?tag=recursion&isCmp=0

(21st July to 31st July):

*Hashing

*Greedy approaches

Hashing:

Resources:

Video: https://www.youtube.com/watch?v=mFY0J5W8Udk

Theory:https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/tutorial/

For more details, refer GeeksForGeeks

Practice problems: (Somewhat sorted as per difficulty level)

1. **https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/perfect-pair-df920e90/**
2. **https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/bob-and-string-easy/**
3. **https://www.geeksforgeeks.org/maximum-distance-two-occurrences-element-array/**
4. **https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/highest-rating-f8ead57a/**
5. **https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/icpc-team-management/**
6. **https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/yet-another-valentines-proposal/**

Greedy algorithm:

Resources:

Video:

1.https://www.youtube.com/watch?v=HzeK7g8cD0Y&list=PLqM7alHXFySESatj68JKWHRVhoJ1BxtLW

2. https://www.youtube.com/watch?v=ARvQcqJ_-NY

Coding problems:

Must-do(For understanding concept):

1.Activity Selection problem (https://www.geeksforgeeks.org/activity-selection-problem-greedy-algo-1/)

2.Job-Sequencing problem (https://www.geeksforgeeks.org/job-sequencing-problem/)

3.Fractional-Knapsack problem

(https://www.geeksforgeeks.org/fractional-knapsack-problem/)

Practice problems:

1. https://www.geeksforgeeks.org/minimum-number-platforms-required-railwaybus-station/#:~:text=We%20are%20given%20two%20arrays,times%20of%20trains%20that%20stop.&text=Explanation%3A%20Only%20one%20platform%20is%20needed.
2. https://www.geeksforgeeks.org/minimize-the-maximum-difference-between-the-heights/
3. https://www.hackerearth.com/practice/algorithms/greedy/basics-of-greedy-algorithms/practice-problems/algorithm/bob-and-walls/
4. Huffman Coding (Only for those who have studied binary trees earlier)

(1st August to 7th August):

-->Revising and practicing on previous topics

(8th August to 14th August):

Dynamic Programming:

Resources:

Intro to DP: **https://www.youtube.com/watch?v=5dRGRueKU3M&list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O&index=46**

Matrix chain Multiplication: **https://www.youtube.com/watch?v=prx1psByp7U&list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O&index=50**

LCS:

https://www.youtube.com/watch?v=sSno9rV8Rhg&list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O&index=60

0-1 Knapsack:

https://www.youtube.com/watch?v=nLmhmB6NzcM&list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O&index=54

Text Resources:
From below link, read mentioned topics and try mentioned problems.
https://www.geeksforgeeks.org/dynamic-programming/
Topics to study: All 4 links of basic concepts
Practice Problems: From basic problems,solve:
Q.No: 2,9,11,18,32,34

(15th August onwards):

Graph Theory:

This is a playlist for Graph theory tutorials. It is very vast and contains tutorials as well as practice problems..

https://www.youtube.com/watch?v=VW85xQ6GJP4&list=PL2q4fbVm1Ik6DCzm9XZJbNwyHtHGclcEh