DSA through C++

Assignment-1: Array Data Structure

1. Define a class Array to implement array data structure with member variables to store capacity of array, last index of the last filled block of the array and a pointer to hold the address of the first block of the dynamically created array.
2. In question 1, define a parameterised constructor to create an array of specified size.
3. In the question 1, add a method to check whether an array is empty or not by returning True or False.
4. In question 1, define a method to append a new element in the array
5. In question 1, define a method to insert a new element at specified index
6. In question 1, define a method to edit an element at specified index.
7. In question 1, define a method to delete an element at specified index.
8. In question 1, define a method to check if the array is full by returning true or false.
9. In question 1, define a method to get element at specified index.
10. In question 1, define a method to count number of elements present in the array.
11. In question 1, define a destructor to deallocate the memory of array.
12. In question 1, define a method to find an element in the array. Return index if the element found, otherwise return -1.

DSA through C++

Assignment-2: array

1. Define a copy constructor in Array class to perform deep copy.
2. Define a copy assignment operator in Array class to perform deep copy.

DSA through C++

Assignment-3: Dynamic Arrays

1. Define a class DynArray to implement dynamic array data structure with member variables to store capacity of array, last index of the last filled block of the array and a pointer to hold the address of the first block of the dynamically created array.
2. In question 1, define a parameterised constructor to create an array of specified size.
3. In question 1, define a method doubleArray() to increase the size of the array by double of its size.
4. In question 1, define a method halfArray() to decrease the size of the array by half of its size.
5. In question 1, define a method which returns the current capacity of the array.
6. In the question 1, add a method to check whether an array is empty or not by returning True or False.
7. In question 1, define a method to append a new element in the array
8. In question 1, define a method to insert a new element at specified index
9. In question 1, define a method to edit an element at specified index.
10. In question 1, define a method to delete an element at specified index.
11. In question 1, define a method to check if the array is full by returning true or false.
12. In question 1, define a method to get element at specified index.
13. In question 1, define a method to count number of elements present in the array.
14. In question 1, define a destructor to deallocate the memory of array.
15. In question 1, define a method to find an element in the array. Return index if the element found, otherwise return -1.

DSA through C++

Assignment-4: Singly Linked List

1. Define a class SLL to implement singly linked list data structure with member variable start pointer of type node.
2. In question 1, define a constructor to initialise start pointer with NULL.
3. In question 1, define a method to insert a data into the list at the beginning.
4. In question 1, define a method to insert a data into the list at the end
5. In question 1, define a method to search a node with the give item.
6. In question 1, define a method to insert a data into the list after the specified node of the list.
7. In question 1, define a method to delete first node from the list.
8. In question 1, define a method to delete last node of the list.
9. In question 1, define a method to delete a specific node.
10. In question 1, define a destructor to deallocates memory for all the nodes in the linked list.

DSA through C++

Assignment-5: Doubly Linked List

1. Define a class DLL to implement doubly linked list data structure with member variable start pointer of type node.
2. In question 1, define a constructor to initialise start pointer with NULL.
3. In question 1, define a method to insert a data into the list at the beginning.
4. In question 1, define a method to insert a data into the list at the end
5. In question 1, define a method to search a node with the give item.
6. In question 1, define a method to insert a data into the list after the specified node of the list.
7. In question 1, define a method to delete first node from the list.
8. In question 1, define a method to delete last node of the list.
9. In question 1, define a method to delete a specific node.
10. In question 1, define a destructor to deallocates memory for all the nodes in the linked list.

DSA through C++

Assignment-6: Circular Linked List

1. Define a class CLL to implement Circular linked list data structure with member variable last pointer of type node.
2. In question 1, define a constructor to initialise last pointer with NULL.
3. In question 1, define a method to insert a data into the list at the beginning.
4. In question 1, define a method to insert a data into the list at the end
5. In question 1, define a method to search a node with the give item.
6. In question 1, define a method to insert a data into the list after the specified node of the list.
7. In question 1, define a method to delete first node from the list.
8. In question 1, define a method to delete last node of the list.
9. In question 1, define a method to delete a specific node.
10. In question 1, define a destructor to deallocates memory for all the nodes in the linked list..

Assignment-7: Circular Doubly Linked List

1. Define a class CDLL to implement Circular Doubly linked list data structure with member variable start pointer of type node.
2. In question 1, define a constructor to initialise start pointer with NULL.
3. In question 1, define a method to insert a data into the list at the beginning.
4. In question 1, define a method to insert a data into the list at the end
5. In question 1, define a method to search a node with the give item.
6. In question 1, define a method to insert a data into the list after the specified node of the list.
7. In question 1, define a method to delete first node from the list.
8. In question 1, define a method to delete last node of the list.
9. In question 1, define a method to delete a specific node.
10. In question 1, define a destructor to deallocates memory for all the nodes in the linked list.

DSA through C++

Assignment-8: Stack using arrays

1. Define a class Stack with capacity, top and ptr pointer as member variables. Implement stack using array.
2. In question 1, define a parameterzied constructor to initialise member variables.
3. In question 1, define a method to push a new element on to the Stack.
4. In question 1, define a method to peek top element of the stack.
5. In question 1, define a method to pop the top element of the stack.
6. In question 1, define a destructor to deallocates the memory.
7. In question 1, define a method to check stack overflow
8. In question 1, define a method to check stack underflow.
9. Define a method to reverse a stack.
10. Define a solution to keep track of minimum value element in the stack.

DSA through C++

Assignment-9: Stack using linked list

1. Define a class Stack with node type pointer **top** as member variable. Implement stack using linked list.
2. In question 1, define a constructor to initialise member variable.
3. In question 1, define a method to push a new element on to the Stack.
4. In question 1, define a method to peek top element of the stack.
5. In question 1, define a method to pop the top element of the stack.
6. In question 1, define a destructor to deallocates the memory.
7. Define a method to reverse a stack.
8. Define a method to check whether a given number is a palindrome or not using stack.
9. Define a method to convert infix to postfix expression.
10. Define a method to evaluate postfix expression.

DSA through C++

Assignment-10: Queue using arrays

1. Define a class Queue with capacity, front, rear and ptr pointer as member variables. Implement queue using array.
2. In question 1, define a parameterzied constructor to initialise member variables.
3. In question 1, define a method to insert a new element at the rear in the queue.
4. In question 1, define a method to view rear element of the queue.
5. In question 1, define a method to view front element of the queue.
6. In question 1, define a method to delete the front element of the queue.
7. In question 1, define a destructor to deallocates the memory.
8. In question 1, define a method to check queue overflow
9. In question 1, define a method to check queue underflow.
10. In question 1, Define a method to count number of elements present in the queue

DSA through C++

Assignment-11: Queue using linked list

1. Define a class Queue with node type pointers front and rear as member variables. Implement queue using Singly linked list.
2. In question 1, define a constructor to initialise member variable.
3. In question 1, define a method to insert a new element at the rear in the queue.
4. In question 1, define a method to view rear element in the queue.
5. In question 1, define a method to view front element in the queue.
6. In question 1, define a method to delete the front element of the queue.
7. In question 1, define a destructor to deallocates the memory.
8. In question 1, define a method to count number of elements present in the queue.

DSA through C++

Assignment-12: Deque

1. Define a class Deque with node type pointers front and rear as member variables. Implement queue using doubly linked list.
2. In question 1, define a constructor to initialise member variables
3. In question 1, define a method to insert a new element at the front
4. In question 1, define a method to insert a new element at the rear.
5. In question 1, define a method to delete front element
6. In question 1, define a method to delete rear element
7. In question 1, define a method to get front element.
8. In question 1, define a method to get rear element
9. In question 1, define a destructor to deallocate the memory.
10. In question 1, define a method to check if deque is empty.

DSA through C++

Assignment-13: Priority Queue

1. Define a class PriorityQueue with node type pointer start as member variable. Implement PriorityQueue using singly linked list.
2. In question 1, define a constructor to initialise member variable.
3. In question 1, define a method to insert new item in the Priority Queue according to the priority number.
4. In question 1, define a method to delete highest priority element
5. In question 1, define a method to get highest priority element
6. In question 1, define a method to get highest priority number.
7. In question 1, define a destructor to deallocate the memory.
8. In question 1, define a method to check if Priority Queue is empty
9. Define a logic to implement priority queue using 2 dimensional arrays
10. Define a logic to implement min priority queue and max priority queue in the same data structure.

DSA through C++

Assignment-14: Tree

1. Define a class BST (Binary Search Tree) with node type pointer root as member variable. Implement Binary Search Tree using linked representation.
2. In question 1, define a constructor to initialise root pointer with NULL.
3. In question 1, define a method to check if the tree is empty.
4. In question 1, define a method to insert a new element in the BST
5. In question 1, define a method for preorder traversing of BST
6. In question 1, define a method for inorder traversing of BST
7. In question 1, define a method for postorder traversing of BST
8. In question 1, define a method to delete an element from BST
9. In question 1, define a method to search an item in the BST
10. In question 1, define a destructor to release memory of all the nodes of BST.

DSA through C++

Assignment-15: Recursion

1. Write a recursive function to print first N natural numbers
2. Write a recursive function to print first N natural numbers in reverse order
3. Write a recursive function to print first N odd natural numbers
4. Write a recursive function to print first N odd natural numbers in reverse order
5. Write a recursive function to print first N even natural numbers
6. Write a recursive function to print first N even natural numbers in reverse order.
7. Write a recursive function to print squares of first N natural numbers
8. Write a recursive function to print squares of first N natural numbers in reverse order.
9. Write a recursive function to print cubes of first N natural numbers
10. Write a recursive function to print cubes of first N natural numbers in reverse order

DSA through C++

Assignment-16: More on Recursion

1. Write a recursive function to calculate sum of first N natural numbers
2. Write a recursive function to calculate sum of first N odd natural numbers
3. Write a recursive function to calculate sum of first N even natural numbers
4. Write a recursive function to calculate sum of squares of first N natural numbers
5. Write a recursive function to calculate factorial of a number
6. Write a recursive function to calculate sum of the digits of a given number
7. Write a recursive function to print binary of a given decimal number
8. Write a recursive function to find nth term of the Fibonacci series
9. Write a recursive function to calculate HCF of two numbers
10. Write a recursive function to calculate x power y.

DSA through C++

Assignment-17: Graph Matrix

1. Define a class Graph using matrix representation with v_count, e_count and adj pointer as instance members.
2. In Question 1, define a method createGraph() to create and store adjacent node information.
3. In question 1, define a method to print graph matrix.
4. In question 1, define a method to print all the adjacent nodes of a given node.
5. In question 1, define a method to check if a given node is isolated node.
6. In question 1, define a destructor to deallocates memory

DSA through C++

Assignment-18: Graph List Representation

1. Define a class Graph to implement linked list representation of graph. Define needful structure for node and class for AdjList.
2. Define appropriate constructors in the classes AdjList and Graph
3. Define appropriate methods to manage linked list in AdjList
4. Define createGraph() method in Graph class to allocate memory for array of AdjList Objects.
5. Define a method addEdge() in Graph class to add a new node in adjacency list.
6. Define destructors in the classes AdjList and Graph
7. Define a method to print graph (print values of adjacency list).

DSA through C++

Assignment-19: Sorting

1. Define a function to implement bubble sort
2. Define a function to implement modified bubble sort to achieve O(n) time complexity in best case.
3. Define a function to implement insertion sort.
4. Define a function to implement selection sort.
5. Define a function to implement quick sort using recursion.
6. Define a function to implement quick sort using iteration.
7. Define a function to implement merge sort using recursion.
8. Define a function to implement merge sort using iteration.
9. Define a class Employee with emp_id, name, salary as instance variables. Provide setters and getters in the class to access instance variables. Also define a function to sort Employee Array data by salary. Use Merge Sort.
10. In question-9, define a function to sort Employee Array data by name. Use Quick Sort.

DSA through C++

Assignment-20: Heap

1. Define a class Heap (implement same as dynamic array).
2. In question-1, define a constructor ti initialise member variables.
3. In question-1, define a method insert() to insert a new element in the heap.
4. In question-1, define a method isEmpty() to check if the heap is empty.
5. In question-1, define a method max() to return greatest value in the heap.
6. In question-1, define a method del() to remove the top element of the heap.
7. In question-1, define a destructor to safely release the memory.
8. In question-1, define a copy constructor to perform deep copy.
9. In question-1, define operator= to perform deep copy.
10. Define a method to sort elements of an array using heap sort.

DSA through C++

Assignment-21: Searching

1. Define a method implementing linear search.
2. Define a method implementing binary search.