

ASSIGNMENT-1 REPORT

General difference between C and C++:

S. No.	C	C++
1.	Does not use classes and objects.	Uses classes and objects
2.	Uses malloc, calloc and free for memory allocation and deallocation.	Uses new and delete for dynamic memory allocation.
3.	Procedural programming language.	Object oriented programming language.
4.	Pointers only, references not allowed and built in data-types only used.	Pointers and references both are available with user defined data-types.
5.	In structures , functions are not allowed.	In structures , functions are also allowed.
6.	Security is less.	Security is more.

ANSWER 1 (Student Records):

S. No.	Similarities:	Differences:
1.	Both have 3 values: id, name, age.	One uses structures and the other uses class for member definition and function.
2.	Both have 2 functions: set, print.	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.
4.	In both the values for number of records are taken at the run time.	

ANSWER 2 (LinkedList):

S. No.	Similarities:	Differences:
1.	Both have 2 members : data, next node (points to the next node in the link list).	One uses structures and the other uses class for member definition and function.
2.	Both have 4 same functions: create, append/insert, delete, display.	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.
4.	Both are dynamic.	

ANSWER 3 (Doubly Linked List):

S. No.	Similarities:	Differences:
1.	Both have 3 members : data, next node (points to the next node in the link list), pre node (points to the previous node of link list).	One uses structures and the other uses class for member definition and function.
2.	Both have 4 same functions: create/insert at beginning, append, delete, display.	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.
4.	Both are Dynamic.	

ANSWER 4 (Circular Link list):

S. No.	Similarities:	Differences:
1.	Both have 2 members : data, next node (points to the next node in the link list) also the last node points to the head or the start node.	One uses structures and the other uses class for member definition and function.
2.	Both have 4 same functions: create, append/insert, delete, display.	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.
4.	Both are dynamic.	

ANSWER 5 (Stack):

S. No.	Similarities:	Differences:
1.	Both have 2 values: element(array), tos(top of stack).	One uses structures and the other uses class for member definition and function.
2.	Both have 3 functions: push, pop, display(to display all the elements of stack).	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.
4.	Both are of static type and size is defined as 10 maximum elements and the same algorithm.	

ANSWER 6 (Queue):

S. No.	Similarities:	Differences:
1.	Both have 3 values: element(array), front, rear (pointers at front and rear of array).	One uses structures and the other uses class for member definition and function.
2.	Both have 3 functions: insert(enqueue), delete(dequeue), display (displaying all elements of queue).	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.
4.	Both are of static type and size is defined as 10 maximum elements and the same algorithm.	

ANSWER 7 (Deque):

S. No.	Similarities:	Differences:
1.	Both have 3 values: element(array), front, rear (pointers at front and rear of array).	One uses structures and the other uses class for member definition and function.
2.	Both have 3 functions: insertatfront(enqueue at front), deleteatfront(dequeue at front), insertatrear(enqueue at rear), deleteatrear(dequeue at rear), display (displaying all elements of queue).	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.
4.	Both are of static type and size is defined as 10 maximum elements and the same algorithm.	

ANSWER 8 (Circular Queue):

S. No.	Similarities:	Differences:
1.	Both have 3 values: element(array), front, rear (pointers at front and rear of array).	One uses structures and the other uses class for member definition and function.
2.	Both have 3 functions: insert(enque), delete(dequeue), display (displaying all elements of queue).	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One is static with 10 elements while other prompts the user for the number of elements
4.	Both use the same algorithm.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.

ANSWER 9 (Priority Queue):

S. No.	Similarities:	Differences:
1.	Both have 3 values: data, priority and order.	One uses structures and the other uses class for member definition and function.
2.	Both have 3 functions: insert(enque), delete(dequeue), display (displaying all elements of queue).	One uses procedural language while the other uses object oriented.
3.	Both use pointers for member access.	One uses malloc and delete, while the other uses new and delete for dynamic memory allocation.