Data Structures and Algorithms

Assignment



**Muhammad Shakaib Arsalan**

**Student ID: F2022266626**

**Course Code: CC2042**

**Section: V16**

**Resource Person: Hafiz Ahsan Arshad**

School of Systems and Technology

UMT Lahore Pakistan

**Flow of Program**

**Class DynamicArray**

**Properties:**

int\* arr

int size

int capacity

**Constructor DynamicArray(capacity)**

Set this.capacity to capacity

Set size to 0

Create an integer array arr of size capacity

**Method getSize()**

Return size

**Method getCapacity()**

Return capacity

**Method insertAtHead(data)**

If size is equal to capacity

Create a new integer array temp of size (capacity + 1)

Copy elements from arr to temp, shifting them to the right

Deallocate memory for arr

Set arr to temp

Increment capacity by 1

Set the first element of arr to data

Increment size by 1

Clear the console screen

Call display(0, 0)

**Method insertAtTail(data)**

If size is equal to capacity

Create a new integer array temp of size (capacity + 1)

Copy elements from arr to temp

Deallocate memory for arr

Set arr to temp

Increment capacity by 1

Set the last element of arr to data

Increment size by 1

Clear the console screen

Call display(size - 1, size - 1)

**Method insertAtPosition(position, data)**

If position is less than 0 or greater than size

Print "Invalid Position"

Return

If size is equal to capacity

Create a new integer array temp of size (capacity + 1)

Copy elements from arr to temp, shifting them to the right

Deallocate memory for arr

Set arr to temp

Increment capacity by 1

Iterate from i=size-1 to position, moving elements to the right

Set arr[position] to data

Increment size by 1

Clear the console screen

Call display(position, position)

**Method deleteFromHead()**

If size is 0

Print "Array is Empty."

Return

Iterate from i=0 to size-2, shifting elements to the left

Decrement size by 1

Print "Done Deletion"

**Method deleteFromTail()**

If size is 0

Print "Array is Empty."

Return

Decrement size by 1

Print "Done Deletion"

**Method deleteFromPosition(position)**

If position is less than 0 or greater than size - 1

Print "Invalid Position"

Return

If size is 0

Print "Array is Empty."

Return

Iterate from i=position to size-2, shifting elements to the left

Decrement size by 1

Print "Done Deletion"

**Method increaseSize(newSize)**

If newSize is less than size

Print "Invalid Size"

Return

Else

Print "Size Increased"

Set capacity to (newSize - 1)

**Method display(start, end)**

Print "Array: "

Iterate over elements in arr

If the index is between start and end (inclusive)

Set text color to green

Print the element

Reset text color

Else

Print the element

**Destructor ~DynamicArray()**

Deallocate memory for arr

**Main Function**

Input num

Initialize DynamicArray object d with num as the initial capacity

Create an integer array arr of size num

If num is greater than 0

Print "Initialize Array"

Iterate from i=0 to num-1

Input arr[i]

Call insertAtTail(arr[i])

While **true**

Print menu of operations

Input choice

**If choice is 1**

Input num

Call insertAtHead(num)

**If choice is 2**

Input num

Call insertAtTail(num)

**If choice is 3**

Input num

Input index

Call insertAtPosition(index, num)

**If choice is 4**

Call deleteFromHead()

Sleep for 1500 milliseconds

Clear the console screen

Call display()

**If choice is 5**

Call deleteFromTail()

Sleep for 1500 milliseconds

Clear the console screen

Call display()

**If choice is 6**

If d.getSize() is greater than 0

Input index

Call deleteFromPosition(index)

Clear the console screen

Else

Print "Array is Empty."

Sleep for 1500 milliseconds

Clear the console screen

Call display()

**If choice is 7**

Clear the console screen

Print "Size of Array is: " + d.getSize()

Print "Capacity of Array: " + d.getCapacity()

Print "Current Values In Array"

Call display()

**If choice is 8**

Input newSize

Call increaseSize(newSize)

Sleep for 1500 milliseconds

Clear the console screen

Call display()

**If choice is 9**

Print "Good Bye :)"

Exit the program

**If choice is not in the range 1 to 9**

Print "Invalid choice"

**Deallocate memory for arr**

**Exit the program**