# Proposed Scheme of Studies for BS in Computer Science (Fall-2020—Spring-2024)Sukkur IBA University - Wikipedia

# Assignment: Data Structures Lab Manual

**Personal Information (First Page):**

- Your Name

- Student ID

- Course Name (Section)

- Instructor Name

- Department Name

- University Name

- Logos of the Department and University (copy from this file) in proper way.

**Table of Contents (Second Page):**

- Contents with Page Numbers

# Topic 1: Arrays

## Task1

|  |
| --- |
| **Problem: Write a program to perform various operations such as creation, insertion, deletion, search and display on single linked list** |
| **Solution:**  printf(“Enter the data: ”); scanf(“%d”, &curr -> data); curr -> link = null;  first = curr;  last = curr; }  void insert() {  int pos, c = 1;  curr=(struct node \*)malloc(sizeof(struct node)); printf(“Enter the data:”);  scanf(“%d”, &curr -> data);  printf(“Enter the position:”);  scanf(“%d”, &pos);  if((pos == 1) && (first != null))  {  curr -> link = first;  first = curr; }  else {  next = first; while(c < pos) {  prev = next;  next = prev -> link;  c++; }  if(prev == null)  {  printf(“\n Invalid position”);  } else {  curr -> link = prev -> link; prev -> link = curr;  if(curr -> link == null) {  last = curr; }  } }  void del() {  int pos, c = 1; printf(“Enter the position”); scanf(“%d”, &pos);  if(first = null) { |

## Task2

|  |
| --- |
| **Problem: Write a program to perform various operations such as creation, insertion, deletion, search and display on single linked list** |
| **Solution:**  printf(“Enter the data: ”); scanf(“%d”, &curr -> data); curr -> link = null;  first = curr;  last = curr; }  void insert() {  int pos, c = 1;  curr=(struct node \*)malloc(sizeof(struct node)); printf(“Enter the data:”);  scanf(“%d”, &curr -> data);  printf(“Enter the position:”);  scanf(“%d”, &pos);  if((pos == 1) && (first != null))  {  curr -> link = first;  first = curr; }  else {  next = first; while(c < pos) {  prev = next;  next = prev -> link;  c++; }  if(prev == null)  {  printf(“\n Invalid position”);  } else {  curr -> link = prev -> link; prev -> link = curr;  if(curr -> link == null) {  last = curr; }  } }  void del() {  int pos, c = 1; printf(“Enter the position”); scanf(“%d”, &pos);  if(first = null) { |

## Task3

|  |
| --- |
| **Problem: Write a program to perform various operations such as creation, insertion, deletion, search and display on single linked list** |
| **Solution:**  printf(“Enter the data: ”); scanf(“%d”, &curr -> data); curr -> link = null;  first = curr;  last = curr; }  void insert() {  int pos, c = 1;  curr=(struct node \*)malloc(sizeof(struct node)); printf(“Enter the data:”);  scanf(“%d”, &curr -> data);  printf(“Enter the position:”);  scanf(“%d”, &pos);  if((pos == 1) && (first != null))  {  curr -> link = first;  first = curr; }  else {  next = first; while(c < pos) {  prev = next;  next = prev -> link;  c++; }  if(prev == null)  {  printf(“\n Invalid position”);  } else {  curr -> link = prev -> link; prev -> link = curr;  if(curr -> link == null) {  last = curr; }  } }  void del() {  int pos, c = 1; printf(“Enter the position”); scanf(“%d”, &pos);  if(first = null) { |

# Topic 2: LinkedList

## Task1

|  |
| --- |
| **Problem: Write a program to perform various operations such as creation, insertion, deletion, search and display on single linked list** |
| **Solution:**  printf(“Enter the data: ”); scanf(“%d”, &curr -> data); curr -> link = null;  first = curr;  last = curr; }  void insert() {  int pos, c = 1;  curr=(struct node \*)malloc(sizeof(struct node)); printf(“Enter the data:”);  scanf(“%d”, &curr -> data);  printf(“Enter the position:”);  scanf(“%d”, &pos);  if((pos == 1) && (first != null))  {  curr -> link = first;  first = curr; }  else {  next = first; while(c < pos) {  prev = next;  next = prev -> link;  c++; }  if(prev == null)  {  printf(“\n Invalid position”);  } else {  curr -> link = prev -> link; prev -> link = curr;  if(curr -> link == null) {  last = curr; }  } }  void del() {  int pos, c = 1; printf(“Enter the position”); scanf(“%d”, &pos);  if(first = null) { |

## Task2

|  |
| --- |
| **Problem: Write a program to perform various operations such as creation, insertion, deletion, search and display on single linked list** |
| **Solution:**  printf(“Enter the data: ”); scanf(“%d”, &curr -> data); curr -> link = null;  first = curr;  last = curr; }  void insert() {  int pos, c = 1;  curr=(struct node \*)malloc(sizeof(struct node)); printf(“Enter the data:”);  scanf(“%d”, &curr -> data);  printf(“Enter the position:”);  scanf(“%d”, &pos);  if((pos == 1) && (first != null))  {  curr -> link = first;  first = curr; }  else {  next = first; while(c < pos) {  prev = next;  next = prev -> link;  c++; }  if(prev == null)  {  printf(“\n Invalid position”);  } else {  curr -> link = prev -> link; prev -> link = curr;  if(curr -> link == null) {  last = curr; }  } }  void del() {  int pos, c = 1; printf(“Enter the position”); scanf(“%d”, &pos);  if(first = null) { |

## Task3

|  |
| --- |
| **Problem: Write a program to perform various operations such as creation, insertion, deletion, search and display on single linked list** |
| **Solution:**  printf(“Enter the data: ”); scanf(“%d”, &curr -> data); curr -> link = null;  first = curr;  last = curr; }  void insert() {  int pos, c = 1;  curr=(struct node \*)malloc(sizeof(struct node)); printf(“Enter the data:”);  scanf(“%d”, &curr -> data);  printf(“Enter the position:”);  scanf(“%d”, &pos);  if((pos == 1) && (first != null))  {  curr -> link = first;  first = curr; }  else {  next = first; while(c < pos) {  prev = next;  next = prev -> link;  c++; }  if(prev == null)  {  printf(“\n Invalid position”);  } else {  curr -> link = prev -> link; prev -> link = curr;  if(curr -> link == null) {  last = curr; }  } }  void del() {  int pos, c = 1; printf(“Enter the position”); scanf(“%d”, &pos);  if(first = null) { |

**Submission Guidelines:**

* All the labs are already uploaded on E-Learning, download and attempt.
* Each topic should be started on new page.
* Topic should be given Heading 1 with Calibri Light, 16 font size, Bold
* Each Task should be given Heading 2 with Garamond, 14 font size, Bold
* Each Problem should be in Calibri Body, 12 font size, Bold
* Each Solution (code) should be in Consolas, 10 font size
* Each solution should be either you have already submitted on E-Learning or should be your code.
* Submit two files (word and its PDF version file) with the specified format.
* Each student has to do assignment individually.
* Ensure the correctness of code (text format).
* Use appropriate headings and labels for each task and topic.
* Follow the order of topics and tasks as mentioned in the assignment.
* Plagiarism is strictly prohibited. If caught, zero marks will be assigned for the entire assignment for all involved students.
* No Late submission (via email/WhatsApp/other platform) is allowed.
* This assignment is **OPTIONAL**, if you attempt it, you will get bonus marks in sessional.

**Note:**

* Make sure to replace placeholders (like "Your Name," "Student ID," etc.) with your actual information.
* Provide complete and working code for each task.
* Include any additional explanations or comments if needed for better understanding.