Singly Linked list

~~IBMCS045~~
IBM19CS045
Derek Stanley
Kannath

```
void insert()
{ int ch, pos, i=1;
  struct node *temp, *newnode;
  int item;
  newnode = (struct node *) malloc(sizeof(struct node));
  printf("Enter the data:");
  scanf("%d", &item);      newnode -> data = item;
  printf("Insert at \n1. Front \n2. Back \n3. Desired Position\n Enter
   choice :");
  scanf("%d", &ch);
  switch(ch)
  { case 1 : newnode -> next = head;
             head = newnode;
             printf("Node created \n");
             break;
    case 2 : temp = head;
             while(temp -> next != NULL)
             { temp = temp -> next; }
             temp -> next = newnode
             newnode -> next = NULL;
             printf("Node created.\n");
             break;
    case 3 : printf("Enter the position: ");
             scanf("%d", &pos);
             if(pos < 2)
             { newnode -> next = head;
               head = newnode; }
             else
             { temp = head;
               while(i != pos-1 || temp -> next != NULL)
               { temp = temp -> next; }
               newnode -> next = temp -> next;
               temp -> next = newnode; }
             printf("Node created"); } }
```

Derek

IBNIKOUS
Derek stanley
Kannath

```
void delfun(int ele)
{ struct node *temp, *del = NULL;
  if (head == NULL)
  { printf("Empty List. Can't delete \n"); return; }
  temp = head;
  if(head -> data == ele)
  { head = head -> next;
    return; }
  while(temp -> next != NULL)
  { if( temp -> next -> data == ele)
    { del = temp -> next;
      if(del -> next == NULL)
        temp -> next = NULLL;
      else
        temp -> next = del -> next;
    }
    else
      temp = temp -> next;
  }
  if(del == NULL)
  printf("Element not found in the list \n"); return; }
}
  void display()
{ struct node *ptr = NULL;
  ptr = head;
  if(ptr == NULL)
    printf("Nothing to print \n");
  else
  while(ptr != NULL)
  { printf("%d", ptr -> data);
    ptr = ptr -> next; }
}
```

Derek