

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Nowadays all the work during the admission of the students is done manually by ink and paper which consumes lot of time and much efforts and very slow process. In the modern world of technology, computers and technologies are playing an important role. Therefore, the aim of the project is to design a program for college admission which consumes less time and less efforts.

1.2 OBJECTIVES

The main objectives of this project are to reduce the time consumption during that time admission process.

- It is user friendly
- It consumes less time
- Easy to fill the application form for the students
- It takes less time to fill the application
- Students can view the details of fee structure for their respective courses
- Declares the list of selected students very easily along with their details

1.3 METHODOLOGY

In this project, data structure plays a important role in college admission system. A program is developed with lot of functions and procedures.

- In this project, it gives the choice for the student that wheatear they want to apply for admission or they want to view the fee structure or to view the list of selected students
- If a student applies for admission, the details are accessed and it is stored according to the priority
- Priority is allotted based on the cet rank. Higher the cet rank, higher will be priority (i.e., 100th rank student will have more priority then the 200th rank student)

- Based on this the students are selected and displayed along with their details.

This is how the project deals with the college admission system.

1.4 EXPECTED OUTCOMES

IF A STUDENT WISH TO APPLY FOR ADMISSION:

1. Automobile engineering
2. Biotech engineering
3. Chemical engineering
4. Electrical engineering
5. Electronics and communication engineering
6. Computer science engineering
7. Information science engineering

Enter the Choice for which course you want to apply

1

Enter your name: Harshith

Enter your father name: Devraj

Enter your sslc roll number: 2016195

Enter your pu roll number: 2018456

Enter your sslc percentage: 98

Enter your pu percentage: 95

Enter your caste: General

Enter your state: Karnataka

Enter your cet rank: 1500

YOUR APPLICATION FORM IS APPLIED SUCCESSFULLY!!

THANK YOU!!

1.5 HARDWARE REQUIREMENTS

- Processor: any processor
- Input device: standard keyboard and mouse
- Output device: high resolution monitor

1.6 SOFTWARE REQUIREMENTS

- Operating system windows
- Turbo C++ compiler

CHAPTER 2

DATA STRUCTURE

DATA STRUCTURE

Data structure is a method of collecting, organizing, retrieving, manipulating the data.

CLASSIFICATION OF DATA STRUCTURE:

There are two types of data structure.

- 1) primitive data structure
- 2) non primitive data structure

1) PRIMITIVE DATA STRUCTURE:

It can be directly manipulated by machine instructions.

It is further classified into 3 types:

- a) int
- b) float
- c) char

2) NON-PRIMITIVE DATA STRUCTURE:

It cannot be directly manipulated by machine instructions.

It is also classified into 3 types:

- a) arrays
- b) lists
- c) files

A) ARRAYS:

Arrays is a set of collections of elements of the same data type. The arrays are stored sequentially in memory. In Java, array elements are treated as objects whereas in c they are not treated as objects.

TYPES OF ARRAYS:

- Single dimensional array
- Multi-dimensional array

SINGLE DIMENSIONAL ARRAY

SYNTAX:

```
data_type array_name [array size];
```

Where

Data_type = type of the data (int, char, float)

Array_name = it is name of the array

Array_size = it is size of the array

EXAMPLE:

```
int a [30];
```

Where

Int is the data type

A is the variable name

30 is the size of the array

MULTI DIMENSIONAL ARRAY

SYNTAX:

```
data_type array_name [row_size] [column_size];
```

Where

Data_type = type of the data (int, char, float)

Array_name = it is name of the array

Row_size = it is size of the row

Column_size = it is the size of the column

EXAMPLE:

```
int a [30[30]];
```

Where

Int is the data type

A is the variable name

30 is the size of the row

30 is the size of the column

B) LIST

In non-primitive data structure, lists are further classified into 2 types. They are:

I) linear data structure

II) nonlinear data structure

LINEAR DATA STRUCTURE:

The elements are organized in linear manner; hence we call it as linear data structure. It is further classified into 3 types:

- Stack
- Queue
- Linked list **STACK:**

It is a linear a data structure in which insertion and deletion of elements happens only at one end (top end). It follows last in first out order (LIFO). In stack, new item is added at the top of a stack. Both insertion and deletion operation are performed from one end of the stack

Different operations of stacks are:

- I) push - to insert the element into the stack.
- II) pop - to delete the elements from the stack.
- III) display - to display the elements present in the stack.

Example:

Just imagine that you have a cookies jar. In this jar, you can add one cookie. The process when you add one cookie inside the jar from topmost position is called as push

operation. Similarly, when you remove one cookie from the jar then that process of cookie removal from the topmost position is called pop operation.

The last entered cookie into the cookie jar is the first to be removed from the jar, thus a stack follows a last in first out principle (LIFO).

QUEUE:

It is a linear data structure in which insertion happens at one end and deletion happened at the other end. It follows First in First out order (FIFO).

Different operations of queues are:

- I) insertion - to insert the element into the queue through rear end.
- II) deletion - to delete the element from the queue through front end.
- III) display - to display the elements present in the queue.

TYPES OF QUEUE:

- Linear queue or ordinary queue
- Circular queue
- Double ended queue
- Priority queue **1) LINEAR QUEUE**

Linear queue is a queue in which

insertion is possible at the rear side of the queue (i.e., end of the queue) and deletion is possible at the front side of the queue (i.e., beginning of the queue).

In linear queue, each the nodes are connected to each other nodes in a sequential manner. The first node's pointer is pointing to the value of the second node and second node's pointer is pointing to the third node and so on. But the first node of the queue has no pointer pointing towards it whereas the last node of the queue has no pointer pointing out from it.

2) CIRCULAR QUEUE

In a circular queue, each node is connected to the next node in the sequential manner but the last node's pointer is also connected to the first node's address. Hence, the last node and the first node also gets connected which results a circular link. Therefore, it is called as circular queue.

3) DOUBLE ENDED QUEUE

The double ended queue is also known as dequeue. It is a queue in which insertion and deletion operations are possible at both the ends (i.e., front and rear side) of the queue.

TYPES OF DOUBLE ENDED QUEUE

- Input restricted double ended queue
- Output restricted double ended queue

INPUT RESTRICTED DOUBLE ENDED QUEUE:

It is the queue in which insertion can happen only at one end whereas deletion can happen at both the ends.

OUTPUT RESTRICTED DOUBLE ENDED QUEUE:

It is the queue in which deletion can happen only at one end whereas insertion can happen at both the ends.

4) PRIORITY QUEUE

Priority queue is the queue in which insertion operation and deletion operation happens based on the priority. The deletion Operation is performed in accordance to priority number (i.e. data item which has highest priority is removed first from the queue) and insertion is performed only in the order.

LINKED LIST

Linked List is a linear data structure. Linked List is also like an array but, the Linked List is not stored sequentially in the memory. Each linked list has 2 parts, the data part and the address part. Data part holds the element or data whereas Address part holds the address of the next element in the list, which is called a node.

Size of the linked list is not fixed and data items can be added at any locations in the linked list. The disadvantage of the linked list is that if we require a particular node, then we must traverse from the first node to the particular node that we require.

BASIC OPERATIONS OF LINKED LIST

1. Insertion - to insert the data or information to the node of the linked list.
2. Deletion - to delete the data or information of the node from the linked list.
3. Display - to display the data or information present in the node of the linked list.

TYPES OF LINKED LIST

1. Single linked list
2. Double linked list
3. Circular linked list
4. Header linked list

1) SINGLE LINKED LIST

Single linked list is the linked list which has only one link. In other words, it is a collection of nodes where each node has two parts i.e., data which contains the elements and link to the next node

It can traverse only in one direction. Single linked list consumes less memory. But the disadvantage of the single linked list is that once the pointer is moved forward, we can't go back.

STRUCTURE REPRESENTATION FOR SINGLE LINKED LIST:

```
struct slist
{
    int data; struct
    slist *ptr;
};
```

2) DOUBLE LINKED LIST

Double linked list is linked list which has two links. In other words, it is a collection of nodes where each node has three parts i.e., link to the previous node, data which contains the elements and link to the next node.

Double linked list consumes more memory. It can be traverse in both the direction.

STRUCTURE REPRESENTATION FOR DOUBLE LINKED LIST:

```
struct slist
{
    struct dlist *prev;
    int data;    struct
    dlist *next;
};
```

3) CIRCULAR QUEUE

Circular linked list is a collection of nodes where the last node points back to the first node which means last node contains the address of first node.

CIRCULAR LINKED LIST IS CLASSIFIED INTO TWO TYPES:

- Circular single linked list
- Circular double linked list

4) HEADER LINKED LIST

Header linked list is a linked list which contains collections of nodes along with one extra node which is called as header node. Header node data is used to store some information such as number of nodes in linked list, address of last nodes, etc.

NON-LINEAR DATA STRUCTURE:

The elements are organized in nonlinear manner; hence we call it as nonlinear data structure. It is further classified into 2 types:

- Tree
- Graph **TREE:**

It is a nonlinear data structure which consists of collection of nodes organized in the hierarchical manner.

TREE TERMINOLOGY

- **ROOT NODE:** This is the node which is at zeroth level.
- **INTERNAL NODE:** All the nodes which has child nodes except the root node.
- **EXTERNAL NODE/LEAD NODE:** Nodes which doesn't have child nodes

TYPES OF TREE

1. Binary tree
2. Strictly binary tree
3. Full binary tree
4. Complete binary tree

1) BINARY TREE

Binary tree is a tree which has at most two child nodes.

PROPERTIES OF BINARY TREE:

- If the height of the binary tree is 'h', then maximum number of leaf nodes is ' 2^h ' (i.e., 2 to the power h).
- If 'h' is the height of the tree, then maximum number of nodes in the tree is ' $2^{h+1} - 1$ ' (i.e., 2 to the power h+1 minus of h).

REPRESENTATION OF BINARY TREE

Binary tree can be represented in two ways:

- Array representation
- Linked list representation

2) STRICTLY BINARY TREE

A tree which has zero or two child nodes is known as strictly binary tree.

3) FULL BINARY TREE

It is binary tree in which every node should have to child nodes except leaf node.

4) COMPLETE BINARY TREE

Complete binary tree is a full binary tree till 'h-1' level but in the last level all the nodes are filled from left to right.

GRAPH:

It is a nonlinear data structure which consists of vertices and edges.

GRAPH TERMINOLOGY

- VERTEX: It is a data element which is nothing but a node. (E.g. : A, B,C)
- EDGE: It is the connection between two nodes. (E.g.: AB, BC)
- DIRECTED GRAPH: A graph which contains only directed edges.
- UNDIRECTED GRAPH: A graph which contains only undirected edges.
- MIXED GRAPH: A graph which contains both directed and undirected edges.

REPRESENTATION OF GRAPHS

They are represented in two ways:

- Array representation
- Linked list representation

MEMORY ALLOCATION

There are two types of memory allocation techniques. They are:

- 1) static memory allocation
- 2) dynamic memory allocation

STATIC MEMORY ALLOCATION:

Allocating memory to variables during compilation time.

DRAWBACK OF STATIC MEMORY ALLOCATION:

- Once the memory is allocated, it can't be increased while executing program used in array.

- If the entire allocated memory is not used, then memory will be wasted.

DYNAMIC MEMORY ALLOCATION:

Allocating memory during execution time or run time. In dynamic memory allocation, memory can be increased while executing a program used in a linked list.

DYNAMIC MEMORY TECHNIQUES ARE:

- Malloc ()
- Calloc ()
- Realloc ()
- Free ()

MALLOC ()

Malloc () is also known as memory allocation. It is used to allocate a single block of memory.

SYNTAX:

```
ptr=(caste-type*) malloc(size);
```

EXAMPLE:

```
ptr=(int*) malloc(sizeof(int));
```

CALLOC ()

Calloc () is also known as contiguous allocation. It is used to allocate multiple blocks of memory.

SYNTAX:

```
ptr=(caste-type*)calloc(n,size);
```

EXAMPLE:

```
ptr=(int*) calloc (5,sizeof(int));
```

REALLOC ()

Realloc () is also known as reallocation. It reallocates the memory occupied by malloc () or calloc() functions.

SYNTAX:

```
ptr=(caste-type*) realloc (ptr,newsize);
```

EXAMPLE:

```
ptr=(int*)malloc(15);
```

```
ptr=(int*) realloc(ptr,30);
```

FREE

It is used to deallocate the memory.it frees the dynamic allocated memory.

STRUCTURE

In c program, struct is a collection of different /unsimilar data types.

SYNTAX

```
struct tag_name  
{  
    type mem 1; type  
    mem 2;  
};
```

Where

Tag_name = structure name

Type = data type (int, char)

Mem 1, mem 2 = members of the structure

EXAMPLE

```
struct student  
{
```

```
char name [30]; char  
roll_number[30]; int  
marks;  
};
```

SELF REFERENTIAL STRUCTURE

It is a structure with different types of members, in which at least one member is pointing to itself.

EXAMPLE

```
struct slist  
{  
char name[30]; char  
father_name[30];  
int sslc_roll; int  
pu_roll; Int  
sslc_percent; int  
pu_percent;  
char caste[30];  
char state[30];  
Int priority;  
struct slist *ptr;  
};
```

CHAPTER 3

ALGORITHM

An Algorithm is a step-by-step instructions or procedures for solving the problems. Algorithm forms the basic foundation of writing a program as well as to understand the program.

Before writing the programs, the following has to be known:

- What is the Input required?
- Which are the tasks to be performed
- What should be the output of the program

Step 1: insert all the header files which are required for this project.

Step 2: declare all the variables which are required for this project.

Step 3: create the structure of single linked list for each course and create a link by using linked list.

Step 4: Students can apply for application with the help of insertion function.

a) if the student clicks on the option 1 (i.e., application for admission) then it displays the courses which are available. There are 7 courses available. They are:

1. Automobile engineering
2. Biotech engineering
3. Chemical engineering
4. Electrical engineering
5. Electronics and communication engineering
6. Computer science engineering
7. Information science engineering

b) student can enter their choice

c) it asks the details of the students such as

- Student name
- Father name
- Sslc roll number

- Pu roll number
- Sslc percentage
- Pu Percentage
- Caste
- State
- Cet rank

d) after filling the application, it prints your application form is applied successfully!! Thank you!!

Step 5: Students can know about the fee structure with the help of display of fee function.

- It displays according to the student's choice
- It contains: course name

: Year

: Sem

: Fee

Step 6: Students can know who are selected with the help display function of single linked list where students are selected based on the priority. Higher the rank higher will be priority.

- The list of selected students is displayed according to the student's choice It contains: student name

: father name

: State

: Cet rank

Step 7: End

CHAPTER 4

IMPLIMENTATION

IMPLEMENTATION OF INSERT PART IN THE SINGLE LINKED LIST

Step 1: create a new node with student details and priority.

Step 2: if head == NULL, then head node will be created. If head is already present then new node will be created.

Step 3: before storing the data, it will check for priority.

Step 4: if head has lower priority then it follows step 5 and 6. If head has higher priority than it follows steps from 7.

Step 5: new1->next = head (i.e., new1 of link stores address of head)

Step 6: head = new1

Step 7: set "start" to head of the list

Step 8: use while loop (start ->next!= NULL && start->next->priority > new1->priority)

Until the condition becomes true, it will assign start= start->next; when the condition becomes true cursor comes out of the loop (end of the loop).

Step 9: new1->next= start->next;(i.e., new1 of link stores address of start node). Step

10: start->next= new1; (i.e., start of link stores address of new1 node)

Step 11: end.

CHAPTER 5

RESULTS

```
NEW HORIZON COLLEGE OF ENGINEERING
AUTONOMOUS COLLEGE PERMANENTLY AFFILIATED TO VTU,APPROVED BY AICTE AND UGC
ACCREDITED BY NAAC WITH 'A' GRADE,ACCREDITED BY NBA
press y to proceed
y
```

Fig 5.1: press y to proceed

```
1.APPLICATION FOR ADMISSION
2.LIST OF SELECTED STUDENTS
3.DISPLAY FEE STRUCTURE
enter the choice
1
1.automobile engineering
2.biotech engineering
3.chemical engineering
4.electrical engineering
5.electronics and communication engineering
6.computer science and engineering
7.information science and engineering
enter the choice to which course you have to apply
1
enter the name
ram
enter your father name
prasad
enter your sslc roll number
2333
enter your pu roll number
2333
enter your sslc percentage
95
enter your pu percentage
96
enter your caste
genral
enter your state
ap
enter your cet rank
2000
YOUR APPLICATION FORM IS APPLIED SUCCESSFULLY
THANK YOU!!
press y to continue
y
```

Fig 5.2: Student 1 entering details for application

```

1.APPLICATION FOR ADMISSION
2.LIST OF SELECTED STUDENTS
3.DISPLAY FEE STRUCTURE
enter the choice
1
1.automobile engineering
2.biotech engineering
3.chemical engineering
4.electrical engineering
5.electronics and communication engineering
6.computer science and engineering
7.information science and engineering
enter the choice to which course you have to apply
1
enter the name
sam
enter your father name
guru
enter your sslc roll number
2910
enter your pu roll number
2710
enter your sslc percentage
85
enter your pu percentage
79
enter your caste
general
enter your state
mp
enter your cet rank
2019
                YOUR APPLICATION FORM IS APPLIED SUCCESSFULLY
                THANK YOU!!
press y to continue

```

Fig 5.3: Student 2 entering details for application

```

press y to continue
y
1.APPLICATION FOR ADMISSION
2.LIST OF SELECTED STUDENTS
3.DISPLAY FEE STRUCTURE
enter the choice
2
1.automobile engineering
2.biotech engineering
3.chemical engineering
4.electrical engineering
5.electronics and communication engineering
6.computer science and engineering
7.information science and engineering
enter the choice to display
1
selected students are:
name = ram
father name = prasad
state = ap
cet rank = 2000
name = sam
father name = guru
state = mp
cet rank = 2019
                THANK YOU!!
press y to continue

```

Fig 5.4: Displaying the list of selected students

```
THANK YOU!!  
press y to continue  
y  
1.APPLICATION FOR ADMISSION  
2.LIST OF SELECTED STUDENTS  
3.DISPLAY FEE STRUCTURE  
enter the choice  
3  
1.automobile engineering  
2.biotech engineering  
3.chemical engineering  
4.electrical engineering  
5.electronics and communication engineering  
6.computer science and engineering  
7.information science and engineering  
enter the choice  
1  
course : automobile engineering  
year : 4  
sem : 8  
fee : 45,000  
THANK YOU!!  
press y to continue
```

Fig 5.5: Displaying the fee structure

CHAPTER 6

CONCLUSION

This is a report which deals with the project of college admission system. This project is developed to apply for application and to display the list of selected students by comparing their cet ranking and also to display the fee structure of respective course.

I thank my reviewer MS. SOJA for helping me to complete this project. I have learned many things about data structures especially linked list through this project.