

Array

- * Array is a collection of similar data type elements.
- * It stores the elements in contiguous memory locations.
- * Accessing is very easy in array because it is index based.

Linked List

- * Linked List is a dynamic data structure in which elements are stored in non-contiguous memory locations.
- * The elements in a linked List are linked using pointers.
- * Linked List contains nodes where each node contains a data field and an address field which refers to the next node.

Types of Pointers

Typed Pointers

- * Typed pointer points to a specific type of data.
- * For example integer pointer pointing to integer data.

Untyped Pointers

- * Untyped pointer points to any data type.

Ex:

```
int main()
{
    int *p; (wild pointer)
    int a = 10;
    p = &a; (NOT a wild pointer)
    *p = 12; (Here value of a is changed)
}
```

Null Pointer

* Null pointer is a pointer which points to nothing.

List

* List is mutable.
* An element in a list can be removed or replaced.
* It is less memory efficient than tuple.
* ordered * Allows duplicate elements

Tuple

* TUPLE is immutable.
* An element in a tuple cannot be inserted or deleted.
* It is more memory efficient than list.
* Allows duplicate elements * ordered

Set itself

* Set is mutable and every set element is unique and must be immutable.
* Unordered and unindexed * no duplicate elements

Dictionary

* Dictionary is mutable.
* It contains key-value pairs * ordered
* Unordered and indexed * no duplicate elements

Different Languages in SQL

Data Definition Language (DDL):

- * DDL contains commands which are required to define the data base.

Ex: create, alter, drop, truncate, rename

Data manipulation Language (DML):

- * DML contains commands which are required to manipulate database.

Ex: insert, update, delete

Data control Language (DCL):

- * DCL is used to deal with user permissions and controls.

Ex: commit, rollback, save point, set transaction

Ex: grant, revoke

Do

transaction control language (TCL):

- * TCL is used to deal with the transaction of the database.

Ex: commit, rollback, save point, set transaction

Data query Language (DQL):

- * It is used to get schema relation based on the query passed to it.

Ex: select

Stack:

- * Stack is a linear data structure in which elements can be inserted or deleted only from one side of the list called top.
 - * Stack follows Last in first out principle.
 - * The element inserted at the last is the first element to come out.
 - * Insertion is called push and deletion of element is called pop.
- Ex: Stack of Plates.

Queue:

- * Queue is a linear data structure in which elements can be inserted only from one side of the list called rear and deleted from another side called front.
- * Queue follows first in first out principle.
- * The element inserted at the first is the first element to come out.
- * Insertion of element is called enqueue and deletion of element is called dequeue.

Ex: A queue of people at ticket window

Structure

- * Structure is a user-defined data type that allows to combining data items of different data types in different memory locations.

Union

- * Union is a user-defined data type that allows to combining data items of different data types in same memory location.

Tree

- * Tree is a non-linear data structure defined as a collection of nodes. Nodes represent values and are connected by edges.
- * It contains one root node.

Graph

- * Graph is a non-linear data structure consists of a finite set of nodes and a set of edges connecting them.

Break

- * Break statement is used to exit from the loop.

Continue

- * This is used to skip the rest of the loop statement and starts the next iteration of the loop to take place.

Normalization

- * Normalization is a process of reducing redundancy by organizing the data into multiple tables.

1NF

- * Each attribute of a table must have atomic values.
- * Column names should be unique.

2NF

- * Should be in 1NF
- * Should not have partial dependency.

3NF

- * Should be in 2NF
- * Should not have transitive dependency.

BCNF (Boyce-Codd Normal Form)

- * Should be in 3NF
- * For each functional dependency $(X \rightarrow Y)$, X should be a super key.

4NF

- * Should be in BCNF
- * Should not have multivalued dependency.

Keys

- * Keys are set of attributes that are used to identify the specific row in a table.

Super key

- * It is a combination of all possible attributes from the relation in order to identify the unique records from the relation.

Ex: Student { sid, sPhno, sName, sBranch }

Super key = { sid, sPhno }

Candidate key

- * Candidate key is a minimal Super key that means it is a Super key without redundant attributes.
- * It should not be null.

Primary key

- * It is a key/attribute that comes from the candidate key in order to retrieve the unique records.
- * It should not be null.

Composite key

- * Composite key is also called candidate key that consists of more than one attributes that together uniquely identify an entity.

Foreign key

- * It is a primary key of another table.
- * It establishes relation between two tables.

Static memory allocation

- * memory allocation done at the time of compilation is known as static memory allocation.
- * In static memory allocation, once the memory is allocated, we cannot change the size of the memory.

Dynamic memory allocation

- * memory allocation done at the time of execution is known as dynamic memory allocation.
- * In dynamic memory allocation, when memory is allocated, we can change the size of the memory.

malloc()

- * malloc is used to dynamically allocate a single block of memory with the specified size.

calloc()

- * calloc is used to dynamically allocate multiple blocks of memory of the specified type.

- * It has two parameters.

free()

- * It is used to dynamically de-allocate the memory.

realloc()

- * It is used to resize a block of memory that was previously allocated.

macro

- * macro is a piece of code in a program that is used to define any constant value.

Ex:

```
#include <stdio.h>
#define LIMIT 5
int main()
{
    printf("%d", LIMIT);
    return 0;
}
```

Output: 5

Command line arguments

- * The arguments passed from command line are called command line arguments.
- * These arguments are handled by main() function.

* Ex: int main(int argc char *argv [])

- * First argument is filename and second argument contains total number of arguments.

Scripting languages

- * scripting language is a subset of programming language.
- * scripting languages are interpreted.
- * They will run independently.

Programming languages

- * programming languages are compiled
- * They will run independently.

DBMS

- * DBMS is a system software for creating and managing databases.

- * In DBMS data is stored in the form of files.

- * DBMS is used by single user and not of above.

- * It doesn't support client-server side interaction.

RDBMS

- * It is a type of DBMS that stores data in tables.

- * The software used to store, manage and retrieve data stored in a relational database is called RDBMS.

- * In RDBMS data is stored in the form of tables.

- * RDBMS is used by multiple users.

- * It supports client-server side interaction.

Advantages: Normalization, High Security, data integrity, data accuracy.

Abstraction in DBMS

- * Hiding the implementation but providing the functionality.

- * It has three Phases

- (i) physical
 - (ii) Logical
 - (iii) view

ACID properties

- * To maintain integrity of a database all transactions must obey ACID properties.

- * There are four ACID properties

- (i) Atomicity

- (ii) consistency

- (iii) Isolation

- (iv) Durability

Types of trees

Binary tree which contains

- * A tree whose elements have almost 2 children.

Binary search tree

- * Nodes in the left subtree are less than the root node and nodes in the right subtree are greater than the root node.

AVL tree

- * It is a self-balancing binary search tree where the difference between heights of left and right subtrees cannot be more than one for all nodes.

Heap tree

- * The tree which satisfies the condition $d(\text{key}) \geq \beta(\text{key})$
- * Where β is child node of d .

Self balancing binary search tree

- * The tree that automatically keeps height as small as possible when insertion and deletion operations are performed on tree.

Red-black tree

- * It is a binary search tree
- * Every node is either red or black.
- * If a node is red then both its children are black.

B-tree

- * B-tree keeps data sorted and allows searches, insertions and deletions in logarithmic ~~optimized~~ time.

B+ tree

- * B+ tree are used to store the large amount of data which can not be stored in the main memory.

m-ary tree

- * An m-ary tree is a rooted tree in which each node has more than m children.

C language (Dennis Ritchie 1972)

- * It is mother language because most of compilers and kernels are written in C today.

GitHub

- * GitHub is a git depository hosting service and it provides web based graphical interface.

CISCO

- * It is an American multinational technology conglomerate corporation.

IT Essentials

- * PC hardware and software curriculum.

React JS

* React JS is a free and open source front-end javascript library for building user interfaces.

IOT

* The Internet of Things describes physical objects that are embedded with sensors, processing ability, software and other technologies.

APSSDC

* Andhra Pradesh state skill development corporation is next gen and above age

what are your career goals (or) Aspiration

* my short term goal is to get a job in reputed company where I can utilize my skills and improve my career path.

my long term goal is to be in respectable position in that organization.

Where will you see yourself in coming 5 years

* I look forward to learn new skills and improve my knowledge to advance my career. I see myself as a knowledgeable professional, having an in-depth knowledge of the company and the industry.

likes and dislikes

- * I like playing cricket
- * I love travelling
- * I don't like cutting
- * I don't like cheating

operating system

- * An operating system is a software that communicates with the hardware and allows other programs to run.

various operating systems

- * windows operating system
- * Linux operating system
- * Android mobile operating system
- * ios mobile operating system

microprocessor

- * microprocessor contains the arithmetic logic and control circuitry required to perform the functions of a CPU.

Hardware

- * physical components of a computer or other electronic devices.

Software

- * Software is a set of instructions to tell the computer what to do.

Machine learning

- * Machine learning is a type of artificial intelligence.
- * Machine learning algorithms use historical data as input to predict new output values.

Deep learning

- * Deep learning is a type of machine learning that imitates the way humans gain certain types of knowledge.

Artificial Intelligence

- * It is the simulation of human intelligence processed by machines especially by computer systems.

Cyber Security

- * Cyber Security is the application of technology used to protect system from cyber attacks.

Cloud computing

- * Cloud computing is a technology that uses internet for storing and managing data on remote servers and then access data via internet.

Ex: Google cloud

Block chain

- * Block chain is an anonymous online ledger that uses data structure to simplify the way we transact.

Big data

- * Big data is a set of technologies used to store, analyse and manage the bulk data.

Data mining

- * Data mining is a process of extracting and discovering patterns in large data sets.

IP address

- * Internet protocol address is unique address that identifies a device on the internet or a local network.

Computer networks

- * Internally connected collection of autonomous computers are called computer networks.

Router

- * Router is a device that provides wifi and is typically connected to a modem.

Call by value

- * The value of a variable passed into a function call as actual parameter is called call by value.

Call by reference

- * The address of a variable passed into a function call as actual parameter is called call by reference.

Function

- * Function is a group of statements that together perform a task.

Features of C

- * Procedural language
- * Fast and efficient
- * modularity
- * .

Features of Python (Guido van Rossum - 1991)

- * Interpreted language
- * Object oriented
- * Open-Source
- * Large Standard Library
- * Easy Language

Features of Java (James Gosling - 1995)

- * Object oriented
- * Compiled and interpreted
- * Portable
- * Secure
- * Platform independent

C, Python, JAVA difference

C

- * It is a static programming language
- * Code is executed directly
- * It is platform dependent
- * It uses compiler to translate code into machine language.

- * It contains pointers.
- * No exception handling is possible.
- * It generates .exe and .bak files.
- REVIEWED c++
- * It is a static programming language.
- * Code is executed directly.
- * It is platform dependent.
- * It uses compiler to translate code into machine language.
- * Exception handling is possible.
- * It is an object oriented programming language.

Java

- * It is a dynamic programming language.
- * Code is executed by JVM.
- * It is platform independent.
- * It uses both compiler and interpreter.
- * It generates .class file.
- * It is a pure object oriented programming language.

Python

- * It is a dynamic programming language.
- * Code is interpreted.
- * It is platform independent.
- * Exception handling is possible.
- * It is a scripting language.
- * Open-Source.

Deadlock in Java

- * Deadlock is a situation where two or more threads are blocked forever, waiting for each other.

Thread Synchronization

- * The capability to control the access of multiple threads to any shared resource is called thread synchronization.

Multi-threading

- * multiple threads act at the same time to complete the process.

SQL Queries

Create Table

* create table student(id int, name varchar(20));

~~Drop~~

Drop Table (or) Database

* drop table college;

* drop database Pace;

Truncate

* Truncate table college; (deletes rows permanently)

Insert

* insert into student(id, name) values(2, 'hi');

Delete

* delete from student where name='hi';

Update

* update student set name = "Hello" where id = 2;

Grant

* grant all on ~~employee~~ student to userA;

* grant sname on student to userB;

Revoke

* revoke all on student to

* revoke update on student from userA;

Select

* select * from student;

Alter

* alter table student add gender varchar(20);

* alter table student drop column empname;

* alter table ~~employee~~ drop column empname;

* alter table student rename column id to column ^{old} _{new};

Desc

* desc table student;

Rename

* rename table student to student1;

Commit

* commit; (: it saves data permanently)

Distinct

* select distinct * from student;

Roll back

- * Rollback; (It cancels the saved data)

Save point

- * Savepoint p00; (It saves upto that when at the time of power loss)

Group by

Syntax columnlist

- * Select * from student group by

col1, col2, ..., coln;

- * Select deptname, sum(salary) from student group by deptname;

Having

Syntax columnlist

- * Select * from student group by col having condition;

- * Select deptname, sum(salary) from student group by deptname having deptname = "Sales";

Order by

- * Select * from student order by empname;

(Ascending)

- * Select * from student order by empname desc;

(Descending)

In

- * Select * from stud where Esal in (10000, 20000);

Between

- * Select * from stud where Esal between 20000 and 30000;

Like

- * Select * from student where ename like 'a%'
where (first character match)
- * Select * from student where ename like '%o'
where (last character match)
- * Select * from student where ename like '—a%'
where (second character match)

Aggregate (or) Group Functions

Sum()

- * Select sum(ESalary) from student;

Avg()

- * Select avg(ESalary) from student;

max()

- * Select max(ESalary) from student;

min()

- * Select min(ESalary) from student;

Count()

- * Select count(ESalary) from student;

(displays highest count of records)

- * Select count(*) from student;

It will displays 15.

Ex: sno	name	age
1	a	21
2	b	24
3	c	23
4		25
5		27

Joins

* Joins are used to establish a connection between two or more tables based on matching columns.

Inner join

* Select * from table1 inner join table2 on table1.columnname = table2.columnname;

A	B	C	D
1	a		
2	c		

C	D
1	b
3	d

A	B	C	D
1	a	1	b
2	c	3	d

Left join

* Select * from table1 left join table2 on table1.columnname = table2.columnname;

A	B	C	D
1	a	1	b
2	c	3	d

C	D
1	b
3	d

A	B	C	D
1	a	1	b
2	c	3	d

Right join

* Select * from table1 right join table2 on table1.columnname = table2.columnname;

A	B
1	a
2	c

C	D
1	b
3	d

A	B	C	D
1	a	1	b
NULL	NULL	3	d

Full join

Select * from table1 union Select * from table2
(Combine both left and right joins)

* Select * from table1 full join table2 on table1.columnname = table2.columnname;
table1.columnname = table2.columnname;
table1.columnname = table2.columnname;

A	B
1	a
2	c

C	D
1	b
3	d

A	B	C	D
1	a	1	b
2	c	3	d

NULL NULL 3 d

AWS (Amazon Web Services)

- * AWS offers a set of global cloud-based products, including ~~storage~~ storage, databases, networking, mobile, IoT, developer tools, security and enterprise applications

53

- * The service offered by AWS that provides object storage through a web service interface.

Array, List in Python

- * Array is a collection of similar data type elements.
- * List is a collection of different data type elements.

Types of DBMS

- * Hierarchical database
- * Network database
- * Relational database
- * Object-oriented database

Latest technologies

- * AI & ML
- * ~~Big data~~ Big data analytics
- * Cyber Security
- * ~~Intelligent Process Automation~~ Intelligent Process Automation
- * ~~Blockchain~~ Blockchain
- * ~~Cognitive computing~~ Cognitive computing

Stack in java

- * Java collection framework provides a stack class that models and implements a stack data structure.

Xampp Server

- * Free and open source cross-platform developed by Apache.

PHP

- * Hypertext Preprocessor is a scripting language used to develop dynamic and interactive websites.

HTML

- * Hypertext markup language is the code that is used to structure a web page.

CSS

- * Cascading style sheet is used to style an HTML document.

SQL

- * Structured Query Language is used to communicate with a database and it is the standard language for RDBMS.

Pickling

- * The process of converting Python object hierarchy into a byte stream is called Pickling.

Unpickling

- * The process of converting byte stream into ^{Python} object hierarchy is called, **unpickling**.

HTML Layout

- * HTML Layout is a blueprint used to organize web pages in a well-defined manner.

Ex: `<header>`, `<nav>`, `<section>`, `<footer>`, `<details>`, `<summary>`

Data Types in Python

- * List, Tuple, String, Dictionary, Boolean, Integer, Set.

In built functions in C

- * `abs`, `calloc`, `exit`, `floor`, `free`, `getS`, `malloc`

String Operations

- * `strlen`, `strcpy`, `strcat`, `strcmp`, `strchr`, `strlwr`

Why we use ";" at the end of statement

- * ~~Semicolon~~ Semi colon is a statement terminator.

Decision making statements in C

- * `if`, `if else`, `nested if`, `if-else-if`, `switch`.

Jump statements in C

- * `break`, `continue`.

TCP/IP (Transmission control protocol / Internet protocol)

- * The protocol stack used in communication over the internet and most other computer networks.

- (i) Physical layer
- (ii) Data link layer
- (iii) Network layer
- (iv) Transport layer
- (v) Application layer

Formatting tags in html

- * ``, ``, `<i>`, `<small>`, `<sub>`, `<sup>`.

Class loader in java

- * Java class loader is a part of the Java runtime environment that dynamically loads Java classes into the ~~JVM~~ JVM.

Program without main function in C

- * Using macro that defines main

```
#include <stdio.h>
#define fun main
int func(void)
{
    printf("Hello");
    return 0;
}
```

Output : Hello

Recursion

- * A function calls itself, is called **Recursion**.
- * Local variables and global variables
- * Local variables are declared inside the function.
- * Global variables are declared outside of the function and are accessible to entire program.

without using semi colon print "Hello world"

* `#include <stdio.h>`

```
int main()
{
    if (printf ("Hello world"))
    {
        return 0;
    }
}
```

OS Interconnection Model

- * open systems interconnection model is reference model used to describe how information from software application in one computer moves through a physical medium to the software application in another computer.

- i) physical layer
- ii) Data link layer
- iii) Network Layer
- iv) Transport Layer
- v) Session Layer
- vi) Presentation Layer
- vii) Application Layer

Types of variables in C

- * Local variables
- * Global variables
- * static variables
- * Automatic variables
- * external variables

Types of variables in java

- * Local variables
- * Instance variables
- * static variables

SDLC (Software Development Life cycle)

- * SDLC is a process followed for a software project within a software organization.

- planning
- Analysis
- Design
- Development
- Testing
- Implementation
- Maintenance

IoT in real time scenario

- * Home automation
- * Biometric security systems
- * Smart cars

python in built functions

- * abs(), dict(), dir(), float(), int(), input(),
list(), map(), print(), min(), max()

Cloud

- * The cloud refers to software and services that run on the internet, instead of locally on your computer.

printf() and scanf()

- * printf() function is used for output. It points the given statement to the console.
- * scanf() function is used for input. It reads the input data from the console.

getchar() and getch()

- * getchar() reads single character from keyboard but it does not use any buffer so the entered character is immediately returned without waiting for the enter key.
- * getch() does not display the entered value on the screen and ~~it does not wait for enter key~~
- * getch() reads single character from keyboard and displays immediately on output screen without waiting for enter key.

Storage classes in C

- * Storage classes are used to describe the following features of a variable/function
 - i) auto
 - ii) extern
 - iii) static
 - iv) register

Add two numbers without using third variable

* `#include <stdio.h>`

`int main()`

`{`

`int a=10, b=5;`

`if (b>0)`

`{`

`while (b>0)`

`{`

`a++;`

`b--;`

`}`

`}`

`if (b<0)`

`{`

`while (b<0)`

`{`

`a--;`

`b++;`

`}`

`}`

`printf ("sum=%d", 0);`

`return 0;`

`}`

Output: sum= 15

void main() and int main

* `void main()` indicates that the `main` function will not return any value.

* `int main()` indicates that the `main` function will return integer type data.

* When our program is simple and is not going to terminate before reaching the last line we can use the `void main()`.

* But if we want to terminate using exit() method, then we have to return some integer values. In that situation void main() will not work so it is better to take int main().

why we need data testing

* Data testing helps the developers to find the problem during fixes.

python ide

* most of the professional developers use PyCharm and it has been considered the best IDE for Python developers.

process and thread difference

* Process is a heavy weight process, ~~process means only program in execution~~.

* Process takes more time to terminate and does not share memory with any other process.

* Thread is a light weight process.

* Thread takes less time to terminate and share memory with other threads.

void main()

* It is the starting point of execution.

* It does not return anything.

class and interface difference

* Class can be instantiated.

* Class does not support multiple inheritance.

* Interface cannot be instantiated.

* Interface supports multiple inheritance.

IaaS, PaaS and SaaS in cloud computing

- * IaaS are cloud-based services, Pay-as-you go for services such as storage, networking and virtualization.
- * PaaS are hardware and software tools available over the internet.
- * SaaS is a software that is available via a third-party over the internet.

monkey patching

- * Monkey patch refers to dynamic modification of a class.
- * We can ~~not~~ change actual behavior of code at runtime.

why should I hire you

- * I have good technical and communication skills. As a fresher I am flexible and adaptive to learn new things. I am sure I will be able to contribute something capable for the growth of the company.

How you overcome your weakness

- * Using my strengths and setting a specific goal I will overcome my weakness.

biggest achievement in life

- * School topper in 10th class. That is my biggest achievement in life.

Tell me about your college

- * My college name is Pace Institute of Technology & Sciences in Angole. Our college provides a very good library and the faculty always motivate us to work hard.

Motivation for work

- * In order to get success I will motivate myself to work hard.

Your role model

- * My role model is Virat Kohli because of his never give up attitude in any difficult situations.

Biggest Fear

- * When I was in work I never care about food, sleep. It may cause health problems so that is my biggest fear. But in order to get success we should overcome our fear. So I will never care about my fear.

Challenges or problems faced while doing your project

- * We did our project on machine learning. As it was a new platform and it took some time in gathering information about machine learning and interact with my teammates and finally done the project.

- * One of my teammates affected by corona so as a team lead I took the responsibility and done his work.

What is the new thing you have learnt in your life?

* Knowledge is ~~very~~ important but in order to get success utilization of knowledge in a proper way is very important.

Self details

* Good morning sir, thank you for giving me this opportunity to introduce myself. My name is Elisha. I am from Ongole. Currently I am pursuing B.Tech final year with CSE from Pace Institute of Technology and Sciences. I completed my Intermediate at Sri Chaitanya Junior College Ongole. I did my schooling in Ongole. There are four members in my family including me. My father is a ... and my ... is a home maker. I have one elder sister. She got married. My strength is my confidence and I am good with people and always listen before I talk. Coming to my weakness when I saw some poor people I am emotional that bothers me. My hobbies are playing cricket and participating in seminars. And my goal is to work in a good MNC company.

Project

* I have done project on medical cost prediction using machine learning regression algorithms. Health care costs increases day by day. As there are greater number of new viruses entering into people, there is a need to predict health charges. This type of prediction helps the government to make a decision regarding health issues.

* We collect the dataset from kaggle, which contains 1338 rows of data with the features age, gender, smoker, BMI, children, region, insurance charges. We need to predict the value of dependent variable using independent variables. So that's why we applied various regression algorithms such as Linear regression, Support vector regression, decision tree regression and random forest regression on this data set to predict medical cost.

* Anaconda navigator is a tool used in this project and we use python language to implement the code. Here we are using sklearn library it is the most useful library for machine learning in python.

```
from sklearn.linear_model import LinearRegression  
obj = LinearRegression()  
obj.fit(xtrain, ytrain)  
y_predict = obj.predict(xtest)
```

```
print("min max score is", r2_score(y_predict, ytest))
```

Output

$r2_score$ is 0.6071988634937787

↓ * Used to evaluate the performance of a linear regression model.

- * It is the percentage of test set tuples.
- * $xtrain$ is the training data set which contains independent variables. 60% data will be used to fit the model and rest 40% will be used to test the model.
- * $xtest$ takes remaining 40% of independent variables and test the used to make predictions to test the accuracy of the model.
- * $ytrain$ is our dependent variable which needs to be predicted by this model.
- * $ytest$ is used to test the accuracy between actual and predicted categories.

collect data set from kaggle

↓
split data into training set and testing set

↓
Apply various regression algorithms

↓
Apply model on testing data

↓
Compare on sheets

↓
predict the output

System Requirements

Software Requirements

- * operating system : windows 7, 8 and 10 (32 and 64 bit)
- * python IDLE / Anaconda Navigator

Hardware Requirements

- * processor - Dual core
- * Speed - 3.1 GHz
- * RAM - 4 GB
- * Hard disk - 200 GB

What is your role in project

- * My role in the project is a team lead and
- o took the responsibility to complete the project on time with perfection.

Project System

System Modules

- * Data collection
- * Data Pre-Processing
- * Modelling
- * Metrics Evaluation

R² Score

- * used to evaluate a performance of a linear regression model

$$R^2 = 1 - \frac{\text{sum of the first errors}}{\text{sum of the second errors}}$$

- * If R² score is 1, then the data is perfectly fit to the model.

Null Pointer Exception

- * Null Pointer Exception is an unchecked exception and extends RuntimeException class.

String buffer

- * StringBuffer is synchronized it means two threads can't call the methods of StringBuffer simultaneously.

- * It is less efficient than StringBuilder

- * It was introduced in Java 1.0

String Builder

- * StringBuilder is non-synchronized. It means two threads can call the methods of StringBuilder simultaneously.

- * It is more efficient than StringBuffer.

- * StringBuilder was introduced in Java 1.5

Types of CSS

Internal CSS

- * It should be written in head between head tags in an html document.

Inline CSS

- * It should be written in between body tags in html document

External CSS

- * External CSS contains separate CSS file which contains only style properties.

HTTP

- * HTTP is unsecured
- * It uses port 80
- * It operates at application layer
- * No SSL certificates are required

HTTPS

- * HTTPS is secured
- * It uses port 443
- * It operates at transport layer
- * SSL certificates are required

PHP

- * Python Enhancement Proposal is a design document providing information to the Python community or describing a new feature for Python

Java API

- * Java API is set of classes included with the Java development environment.

Python Scraping

- * It is the process of collecting and parsing raw data from the web.

Binary Search

- * The input to binary search must be in ascending order.

- * First, key element is compared with middle element of array.

- * If key element is equal to middle element of array, then successful search.

- * If key element is less than the middle element of array then search in left part.
- * If key element is greater than middle element of array then search in right part.

Bubble Sort

- * In bubble sort, in each iteration we compare adjacent elements.
- * If an element is greater then swap elements.
- * After first iteration the biggest element is moved to the last position.
- * After second iteration, the next biggest element is moved to ~~nexts~~ last ~~but~~ one ~~position~~.
- * This process continues until the array gets sorted.
- * Here we require ~~one~~ 2 passes.

Insertion Sort

In first step

- * The second element of an array is compared with the elements that appears before it. If the second element is smallest than first element then second element is inserted in the position of first element.
- * After first step first two elements of an array will be sorted.
- * After second step first three elements of an array will be sorted.

Selection Sort

- * In Selection Sort first find the smallest element in the array and swap with 0th position element. So, after 1st iteration the smallest element will be at 0th position.
- * Then find the second smallest element in the array and swap with 1st position element. So, after 2nd iteration the second smallest element will be at 1st position.
- * Repeat this process until array is sorted.
- * This process continues until the array gets sorted.

Merge Sort

- * Merge sort works on the principle of divide and conquer technique.
- * The given array is divided into 2 ~~parts~~ ^{recursively}.
- * Each subarray is sorted ~~recursively~~ so that the first subarray and second subarray are in sorted order.
- * Merge the solutions of 2 sub arrays.

Quick Sort

- * Quick sort is implemented based on the principle of divide and conquer.
- * Select the first element of an array as pivot element. Divide the given array

into 2 parts such that the left part contains all the elements which are less than the pivot element and right part contains all the elements which are greater than the pivot element. This process is called Partitioning.

- * The left part and right part will be sorted recursively, so that the entire array will be in sorted order.
- * There is no need of combine in quick sort.

Single linked List operations

* Void append C)

{

```
struct node* temp*P;
temp=(struct node*) malloc(sizeof(struct node));
printf("enter data");
scanf("%d",temp->data);
temp->link=NULL;
if (root==NULL)
    root=temp;
else
{
    p=root;
    while (p->link!=NULL)
        p=p->link;
    p->link=temp;
}
```

3

4

```
* void addatbegin()
{
    struct node* temp;
    temp=(struct node*) malloc(sizeof(struct node));
    printf("enter data");
    scanf("%d", &temp->data);
    temp->link=NULL;
    if (root==NULL)
        root=temp;
    else
    {
        temp->link=root;
        root=temp;
    }
}
```

```
* void addatloc()
{
    int loc;
    printf("enter location");
    scanf("%d", &loc);
    len=length();
    if (loc>len)
        printf("Invalid location\n");
    else
    {
        struct node* temp, *p=root;
        int i=1;
        temp=(struct node*) malloc(sizeof(struct node));
        temp->data=
```

```

    printf("Enter data");
    scanf("%d", &temp->data);

    temp->link = NULL;
    while(i < loc)
    {
        p = p->link;
        i++;
    }
    temp->link = p->link;
    p->link = temp;
}

*int length()
{
    int count = 0;
    struct node* temp;
    temp = root;
    while(temp != NULL)
    {
        count++;
        temp = temp->link;
    }
    return count;
}

*void display()
{
    struct node* temp;
    temp = root;
    if(root == NULL)
        printf("list is empty\n");
}

```

```

    else
    {
        while (temp != NULL)
        {
            printf("%d", temp->data);
            temp = temp->link;
        }
    }
}

```

*void delPosC()

```

{
    int loc;
    printf("enter location");
    scanf("%d", &loc);
    if (loc > length())
        printf("invalid location");
    else if (loc == 1)
    {
        struct node *temp;
        temp = root;
        root = temp->link;
        temp->link = NULL;
    }
    else
    {
        struct node *temp;
        temp = root;
        int i = 1;

```

→ void delAStart()

temp = q->next

temp->p = p->next

q->next = p->next

temp = q->next

temp->p = p->next

temp = q->next

temp->p = p->next

temp = q->next

temp->p = p->next

```

while (i < loc-1) {
    p = p->link;
    i++;
}
temp = p->link;
p->link = temp->link;
temp->link = NULL;
}

```

del at pos
del after
del at before or after

* void Reverselist()

```

int i, j, len;
struct node *p, *q, *temp;
len = length();
i = 0;
j = len - 1;
p = q = root;
while (i < j) {
    k = 0;
    while (k < j)
        {
            q = q->link;
            k++;
        }
    temp = p->data;
    p->data = q->data;
    q->data = temp;
    i++;
    j--;
    p = p->link;
    q = root;
}

```

* void swap()

```

int i = 1;
struct node *p = root, *q, *r;
printf("enter location (n)");
scanf("%d", &loc);
while (i < loc-1) {
    p = p->link;
    i = i + 1;
}
q = p->link;
r = q->link;
q->link = r->link;
r->link = q;
p->link = r;
}

```

fibonacci

```
def fibo(n):  
    n1=0, n2=1, n3=0  
    if n>0:  
        n3=n1+n2;  
        n1=n2;  
        n2=n3;  
        print(n3)  
        fibo(n-1)
```

```
n=int(input())  
print(0,1)  
print(fibo(n-2))
```

palindrome

```
while(n>0):  
    r=n%10  
    sum=sum*10+r  
    n=n/10  
if (temp==sum):  
    print("Palindrome")
```

Factorial

```
def fact(n):  
    if n==0:  
        return 1  
    else:  
        return (n*fact(n-1))  
  
n=int(input())  
print(fact(n))
```

fibonacci

```
def fibo(n):  
    if n<=1:  
        return n  
    else:  
        return (fibo(n-1)+fibo(n-2))  
  
n=int(input())  
if n<=0:  
    print("Not valid")  
else:  
    for i in range(n):  
        print(fibo(i))
```

Stack operations

```
* void pushC()
{
    if (top == size-1)
        printf("Stack overflow");
    else
        printf("Enter element");
    scanf("%d", &ele);
    top++;
    stack[top] = ele;
}
```

* void popC()

```

if (top == -1)
{
    printf("The stack is Underflow");
}
else
{
    ele = stack[top];
    top--;
    printf("The deleted element %d", ele);
}
```

```
* void display()
{
    int i;
    if (top == -1)
    {
        printf(" stack is empty");
    }
    else
    {
        for (i = top; i >= 0; i--)
            printf("In %d", stack[i]);
    }
}
```

Queue operations

```
* void enqueue()
{
    if (r == size - 1)
        printf("In queue overflow");
    else
    {
        printf("In enter element");
        scanf("%d", &ele);
        r++;
        Q[r] = ele;
    }
    if (f == -1)
    {
        f++;
    }
}
```

*void dequeue()

{

if ($f = -1$)

printf("In Queue Underflow");

else

{

ele = q[f];

~~if (f == -1)~~ if ($f = -1$)

{

~~f = -1~~

~~r = -1~~

}

else

{

printf("Deleted element %d", ele);

$f = f + 1$;

}

}

*void display()

{

if ($f = -1$)

printf("Queue empty");

else

{

printf("Elements in Queue are ");

for (i = f; i <= r; i++)

{

printf(" %d", q[i]);

}

}

}

Binary Search program

```
flag=0;  
low=0;  
high=n-1;  
while (low <= high)  
{  
    mid = (low+high)/2;  
    if (key == a[mid])  
    {  
        flag=1;  
        break;  
    }  
    else if (key < a[mid])  
    {  
        high = mid-1;  
    }  
    else  
    {  
        low = mid+1;  
    }  
}  
if (flag == 1)  
    printf("Key found");  
else  
    printf("Not found");  
getch();  
}
```

Bubble Sort program

```
for (i=0; i < n; i++)  
{  
    for (j=0; j < n-1; j++)  
    {  
        if (a[i] > a[i+1])  
        {  
            temp = a[i];  
            a[i] = a[i+1];  
            a[i+1] = temp;  
        }  
    }  
}
```

Insertion Sort program

```
for(i=1; i<n; i++)  
{  
    temp = a[i];  
    for(j=i; j>0 && a[j-1]>temp; j--)  
    {  
        a[i] = temp;  
    }  
}
```

Selection Sort Program

```
for(i=0; i<n; i++)  
{  
    min = i;  
    for(j=i+1; j<n; j++)  
    {  
        if(a[j] < a[min])  
            min = j;  
    }  
    temp = a[i];  
    a[i] = a[min];  
    a[min] = temp;  
}
```

final

* final is the keyword and access modifiers which is used to apply restrictions on a class, method or variable.

finally

* finally is the block in java exception handling to execute the important code whether the exception occurs or not.

Insertion Sort program

```
for(i=1; i<n; i++)  
{  
    key = a[i];  
    j = i-1;  
    while(j>=0 && a[j]>key)  
    {  
        a[j+1] = a[j];  
        j = j-1;  
    }  
    a[j+1] = key;  
}
```

finalize

* finalize is the method in java which is used to perform clean up processing just before object is garbage collected.

Local variables

- * A variable defined within a block or method or constructor is called a local variable.
- * These variables are created when the function is called and destroyed after when the call returns from the function.
- * Initialization is mandatory before using it.

Instance variables

- * Instance variables are not static variables and are declared in a class outside the method.
- * These variables are created when an object of the class is created and destroyed when the object is destroyed.
- * Its default value is zero.

Static variables

- * Static variables are declared using the static keyword within a class outside the method.
- * Its default value is zero.
- * It is a common property to all the objects.

Data Structures

- * Data structures are used to store data in an organised and efficient manner.

Time complexities of Sorting algorithms

* Bubble Sort

Best - $O(n)$

Average - $O(n^2)$

Worst - $O(n^2)$

* Insertion Sort

Best - $O(n)$

Average - $O(n^2)$

Worst - $O(n^2)$

* Selection Sort

Best - $O(n^2)$

Average - $O(n^2)$

Worst - $O(n^2)$

* Merge Sort

Best - $O(n \log n)$

Average - $O(n \log n)$

Worst - $O(n \log n)$

* Quick Sort

Best - $O(n \log n)$

Average - $O(n \log n)$

Worst - $O(n^2)$

* Linear Search - $O(n)$

* Binary Search - $O(\log n)$

* Why you choose IT

* Because IT sector is one of the fastest-growing sectors and remains the fastest-growing for us.

* SOFT provides many opportunities for us.

So that's why I choose this.