Basic Concepts of Big Data (20 Hours)

- **Session 1-4:**

o Concept and characteristics of Big Data
o History of Big Data
o Jobs in Big Data
o Types of Big data (structured, semi-structured, unstructured)

- **Session 5-9:**

o Big Data Frameworks
o Big Data Programming Paradigms
o Big Data Programming Languages

- **Session 10-11:**

o Introduction to Data Science and Skillset required for working with Big Data

- **Session 12-15:**

o Simplified Overview of Machine Learning Algorithms and Neural Networks
o Types of Machine Learning (Supervised, Un-Supervised, Reinforcement)

- **Session 16-18:**

ACTS, Head Quarters, Pune
o Examples of Big Data and Data Science in Practice (Healthcare, Logistics & Transportation, Manufacturing etc.

- **Session 19-20:**

o Application Examples and Real –World Use Cases (e.g., Healthcare, finance, marketing, etc.)

Types of Big Data Technologies (+ Management Tools)

1. Data storage

- Apache Hadoop
- Apache Spark
- Apache Hive
- Apache Flume
- ElasticSearch
- MongoDB

2. Data mining

- Rapidminer
- Presto

3. Data analytics

- Apache Spark
- Splunk
- KNIME

4. Data visualization

- Tableau
- Power BI

Apache Hadoop

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

MongoDB

MongoDB is a source-available cross-platform document-oriented database program.

Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas

Rapidminer

pros:

1. Multiple deployment options based on our preference.
2. Strong visualization.
3. Accurate Preprocessing.
4. Multiple interfaces.
5. Java API available that can be used in programs.

cons:

1. It takes too much memory and so slows down your system.
2. Less forums for support.

Presto

A single Presto query can process data from multiple sources like HDFS, MySQL, Cassandra, Hive and many more data sources. Presto is built in Java and easy to integrate with other data infrastructure components. Presto is powerful, and leading companies like Airbnb, DropBox, Groupon, Netflix are adopting it.

Apache Spark

Apache Spark is an open-source unified analytics engine for large-scale data processing. Spark provides an interface for programming clusters with implicit data parallelism and fault tolerance

Splunk

Splunk is a program that enables the search and analysis of computer data. It analyzes semi-structured data and logs generated by various processes with proper data modeling as per the need of the IT companies. The user produces the data by means of any device like- web apps, sensors, or computers. It has built-in functionality for defining data types, field separators, and search process optimization. For the searched result, it also provides visualization of data.

Tableau can handle huge columns of data and still offer better performance.	Power BI is best for a limited volume of data.
Tableau has better data visualization.	Power BI offers many data points for data visualization.
Tableau works best with huge data.	Power BI is suboptimal with huge data.
Experts and experienced users use Tableau.	Power BI is used by beginners and experienced alike.

8. HPCC Systems

1. Airflow

9. Hudi

10. Iceberg

2. Delta Lake

19. MongoDB

3. Drill

11. Kafka

4. Druid

18. Trino

12. Kylin

5. Flink

17. Storm

13. Pinot

6. Hadoop

7. Hive

16. Spark

14. Presto

15. Samza

1. Data storage

- Apache Hadoop
- Hortonworks
- Data lake
- Apache Spark
- MongoDB
- Apache Cassandra
- P r e s t o
- Cloudera
- Cloud storage
- Elastic search
- Hybrid storage

Cloud Service Providers

Microsoft Azure

Google Cloud Platform

Amazon Web Service (AWS)

Oracle Cloud

IBM Cloud Services

Rackspace

Adobe Creative Cloud

Red Hat

SAP

Kamatera

Salesforce

Verizon Cloud

VMware

1. Airflow

Airflow in Apache is a popularly used tool to manage the automation of tasks and their workflows. They are also primarily used for scheduling various tasks. Consider that you are working as a data engineer or an analyst and we might need to continuously repeat a task that needs the same effort and time every time. The kind of such tasks might consist of **extracting, loading,** or **transforming data** that need a regular analytical report. We can simply automate such tasks using Airflow in Apache by training your machine learning model to serve these kinds of tasks on a regular interval specified while training it

2. Delta Lake

What is Big Data

Data which are very large in size is called Big Data. Normally we work on data of size MB(WordDoc ,Excel) or maximum GB(Movies, Codes) but data in Peta bytes i.e. 10^15 byte size is called Big Data. It is stated that almost 90% of today's data has been generated in the past 3 years.

Sources of Big Data

These data come from many sources like

- **Social networking sites:** Facebook, Google, LinkedIn all these sites generates huge amount of data on a day to day basis as they have billions of users worldwide.

- **E-commerce site:** Sites like Amazon, Flipkart, Alibaba generates huge amount of logs from which users buying trends can be traced.

- **Weather Station:** All the weather station and satellite gives very huge data which are stored and manipulated to forecast weather.

- **Telecom company:** Telecom giants like Airtel, Vodafone study the user trends and accordingly publish their plans and for this they store the data of its million users.

- **Share Market:** Stock exchange across the world generates huge amount of data through its daily transaction.

3V's of Big Data

1.Velocity: The data is increasing at a very fast rate. It is estimated that the volume of data will double in every 2 years.

2.Variety: Now a days data are not stored in rows and column. Data is structured as well as unstructured. Log file, CCTV footage is unstructured data. Data which can be saved in tables are structured data like the transaction data of the bank.

3.Volume: The amount of data which we deal with is of very large size of Peta bytes.

Use case

An e-commerce site XYZ (having 100 million users) wants to offer a gift voucher of 100$ to its top 10 customers who have spent the most in the previous year.Moreover, they want to find the buying trend of these customers so that company can suggest more items related to them.

Issues

Huge amount of unstructured data which needs to be stored, processed and analyzed.

Solution

Storage: This huge amount of data, Hadoop uses HDFS (Hadoop Distributed File System) which uses commodity hardware to form clusters and store data in a distributed fashion. It works on Write once, read many times principle.

Processing: Map Reduce paradigm is applied to data distributed over network to find the required output.

Analyze: Pig, Hive can be used to analyze the data.

Cost: Hadoop is open source so the cost is no more an issue.

What is Hadoop

Hadoop is an open source framework from Apache and is used to store process and analyze data which are very huge in volume. Hadoop is written in Java and is not OLAP (online analytical processing). It is used for batch/offline processing.It is being used by Facebook, Yahoo, Google, Twitter, LinkedIn and many more. Moreover it can be scaled up just by adding nodes in the cluster.

Modules of Hadoop

1.HDFS: Hadoop Distributed File System. Google published its paper GFS and on the basis of that HDFS was developed. It states that the files will be broken into blocks and stored in nodes over the distributed architecture.

2.Yarn: Yet another Resource Negotiator is used for job scheduling and manage the cluster.

3.Map Reduce: This is a framework which helps Java programs to do the parallel computation on data using key value pair. The Map task takes input data and converts it into a data set which can be computed in Key value pair. The output of Map task is consumed by reduce task and then the out of reducer gives the desired result.

4.Hadoop Common: These Java libraries are used to start Hadoop and are used by other Hadoop modules.

Hadoop Architecture

The Hadoop architecture is a package of the file system, MapReduce engine and the HDFS (Hadoop Distributed File System). The MapReduce engine can be MapReduce/MR1 or YARN/MR2.

A Hadoop cluster consists of a single master and multiple slave nodes. The master node includes Job Tracker, Task Tracker, NameNode, and DataNode whereas the slave node includes DataNode and TaskTracker.

Using Big Data Analytics Tools

1.Healthcare: Big data analytics technologies and tools are being used in healthcare to predict patient outcomes, identify at-risk patients, and improve population health.

2.Retail: Big data analytics tools are being used by retailers to improve customer experience, target marketing campaigns, and prevent fraud.

3.Manufacturing: Big data analytics tools are being used in manufacturing to improve quality control, reduce downtime, and optimize production processes.

4.Banking: Real time big data analytics tools are being used by banks to detect fraudulent activities, prevent money laundering, and improve customer service.

1.Government: Big data analytics tools are being used by government agencies to improve public services, combat fraud and corruption, and better understand citizen needs.

Limitations of Big Data Analytics Tools

There are several limitations to big data analytics tools, including:

1.They can be expensive and require a lot of resources to implement.

2.They can be complex to use and require skilled staff to get the most out of them.

3.They can require a lot of data to be effective, which can be a challenge to collect.

4.They can be slow and may not be able to keep up with rapidly changing data.

5.They can produce biased results, depending on how they are configured.

Best tool for big data analytics?

Big Data frameworks such as Apache Hadoop are widely used in the market. Clusters of computers can be used to process massive data sets using Hadoop. Scaling up from one server to tens of thousands of commodity computers is one of the best features of this Big Data Tool.

Five types of big data analytics?

The five types of big data analytics are as follows:

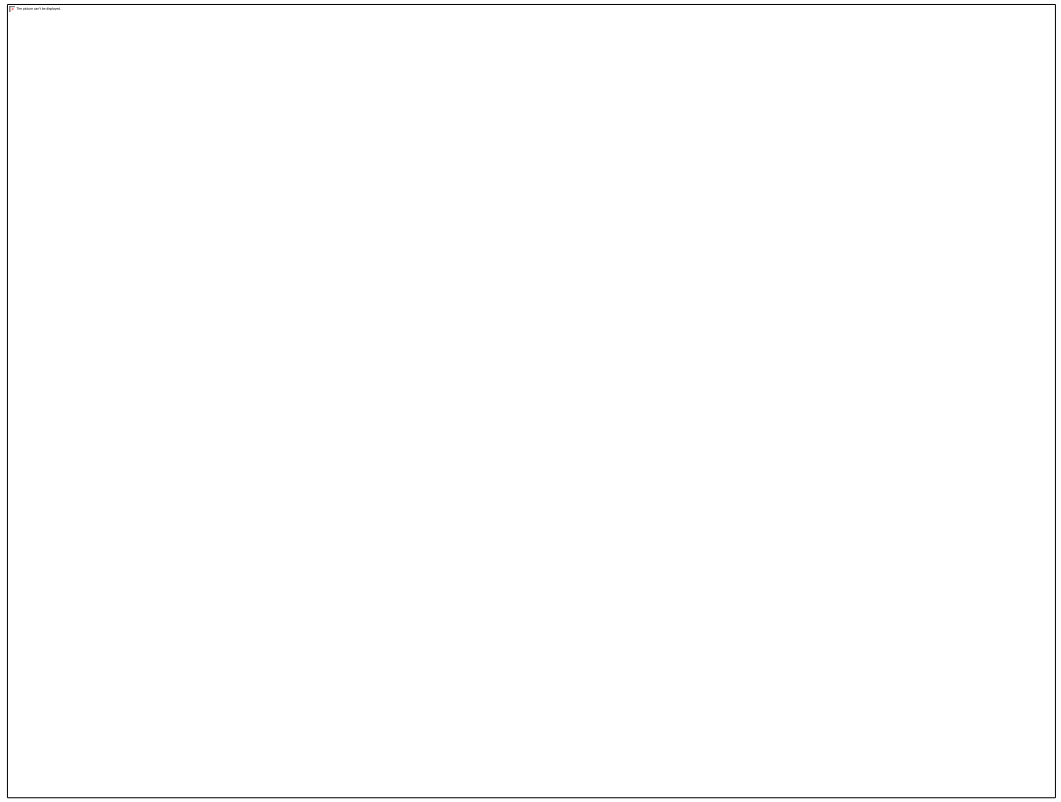
Cyber Analytics
Prescriptive Analytics
Descriptive Analytics
Diagnostic Analytics
Predictive Analytics

Data engineering

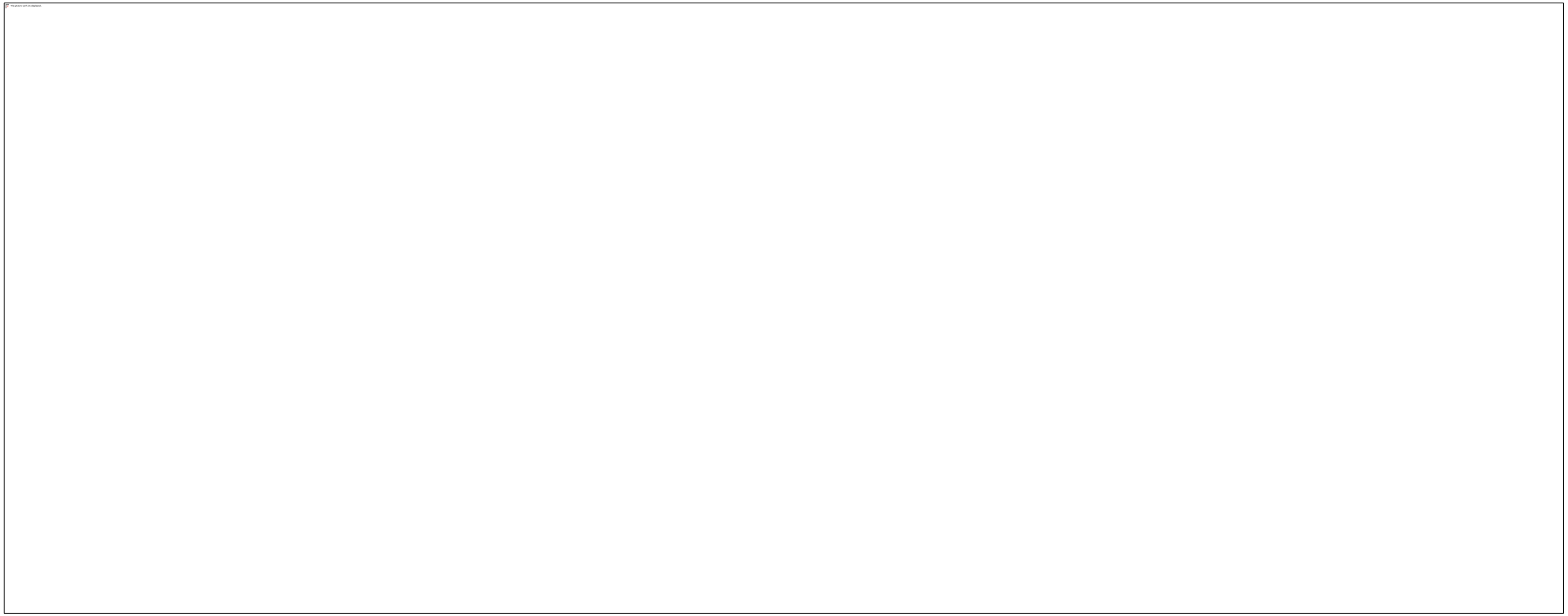
Data engineering is the process of building robust data architecture that allows for data processing. This includes data transfers between databases and building data warehouses for easy accessibility.

Through data engineering, the following question is answered: “How do I make all the data we collect easier for our data analysts and other stakeholders to wade through?” Data engineering makes the data more reliable, accurate, and ingestible through robust data processing systems.

Data Engineer and a Data Analyst?



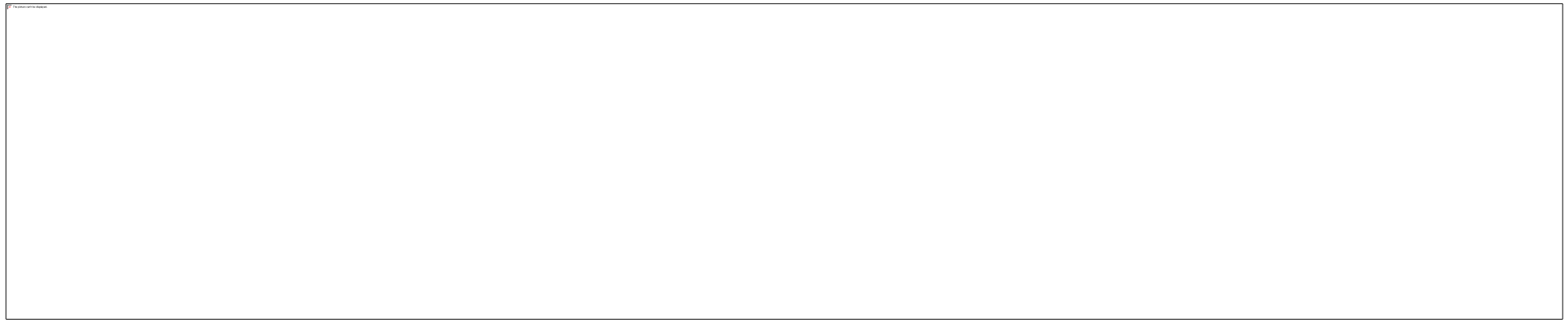
Data Engineering



Requirements To Become a Data Engineer

Requirements To Become a Data Analyst

Data Analyst, Data Engineer and Data Scientist



1. **Data in ____ bytes size is called Big Data.**

Tera
Giga
Peta
Meta

2. **How many V's of Big Data?**

2
3
4
5
Answer: D) 5
Volume, Velocity, Variety, Value and Veracity

3. Unprocessed data or processed data are observations or measurements that can be expressed as text, numbers, or other types of media.

True
False

4. In Big Data environments, Velocity refers –

Data can arrive at fast speed

Enormous datasets can accumulate within very short periods of time

Velocity of data translates into the amount of time it takes for the data to be processed

All of the mentioned above

5. In Big Data environments, Variety of data includes –

Includes multiple formats and types of data

Includes structured data in the form of financial transactions,

Includes semi-structured data in the form of emails and unstructured data in the form of images

All of the mentioned above

6. Which of the following are Benefits of Big Data Processing?

Cost Reduction

Time Reductions

Smarter Business Decisions

All of the mentioned above

7. Data that does not conform to a data model or data schema is known as _____.

Structured data

Unstructured data

Semi-structured data

All of the mentioned above

8. Amongst which of the following is/are not Big Data Technologies?

Apache Hadoop
Apache Spark
Apache Kafka
Apache Pytarch

9._____ involves the simultaneous execution of multiple sub-tasks that collectively comprise a larger task.

Parallel data processing

Single channel processing

Multi data processing

None of the mentioned above

10 Amongst which of the following can be considered as the main source of unstructured data.

Twitter

Facebook

Webpages

All of the mentioned above

11. Amongst which of the following shows an example of unstructured data,

Students roll number, age

Videos

Audio files

Both B and C

12 Scalability, elasticity, resource pooling, self-service, low cost and fault tolerance are the features of,

Cloud computing

Power BI

System development

None of the mentioned above

Big data is a field that treats ways to analyze and systematically extract information from or otherwise deal with data sets. Data can be large or complex to be dealt with by traditional data processing applications software

A large amount of data
It is a popular term used to express the exponential growth of data.
Big data is difficult to store, collect, maintain, analyze and visualize.

Distributed file system: A distributed file system is a file system in which data is stored on a server. The data is accessed and processed as if it were stored on the local client machine. The following are the Characteristics of distributed file system:

Transparency
user mobility
Performance
simplicity and ease of use
Scalability
high availability
high reliability
Big data tools: Apache Hadoop, Apache Storm, Cassandra, Mongo DB, Neo4j. Learn More.

Big data sources:

Amazon, Redshift, Mongo DB

Challenges of big data:

Uncertainty of data management
The talent gap in big data
Getting data into a big data structure
Synchronizing across data sources
Integration

Benefits of big data:

Cost
Time reduction
Speeding up decision-making
Analyze in real-time
Model and Test variation

Characteristics of big data:

Volume
Velocity
Variety

Types of big data:

Structured
unstructured
Semi-structured
hybrid

Use cases of big data:

Recommendation engine
Analyzing call detail records
Fraud detection
sentiment analysis

What is Structured Data?

Structured data is information that has been formatted and transformed into a well-defined data model. The raw data is mapped into predesigned fields that can then be extracted and read through SQL easily. SQL relational databases, consisting of tables with rows and columns, are the perfect example of structured data.

The relational model of this data format utilizes memory since it minimizes data redundancy. However, this also means that structured data is more inter-dependent and less flexible. Now let's look at more examples of structured data.

Examples of Structured Data

This type of data is generated by both humans and machines. There are numerous examples of structured data from machines, such as POS data like quantity, barcodes, and weblog statistics. Similarly, anyone who works on data would have used spreadsheets once in their lifetime, which is a classic case of structured data generated by humans. Due to the organization of structured data, it is easier to analyze than both semi-structured and unstructured data.

What is Semi-Structured Data

We may not always find your data sets to be structured or unstructured. Semi-structured data or partially structured data is another category between structured and unstructured data. Semi-structured data is a type of data that has some consistent and definite characteristics.
It does not confine into a rigid structure such as that needed for relational databases. Businesses use organizational properties like metadata or semantics tags with semi-structured data to make it more manageable. However, it still contains some variability and inconsistency.

Examples of Semi-Structured Data

An example of data in a semi-structured format is delimited files. It contains elements that can break down the data into separate hierarchies. Similarly, in digital photographs, the image does not have a pre-defined structure itself but has certain structural attributes making them semi-structured. F or instance, if you take a photo from a smartphone, it would have some structured attributes like geotag, device ID, and DateTime stamp. After you save them, we can assign tags to images such as ‘pet’ or ‘dog’ to provide a structure. On some occasions, unstructured data is classified as semi-structured data because it has one or more classifying attributes.

What is Unstructured Data?

Unstructured data is defined as data present in absolute raw form. This data is difficult to process due to its complex arrangement and formatting.

Unstructured data includes social media posts, chats, satellite imagery, IoT sensor data, emails, and presentations. Unstructured data management takes this data to organize it in a logical, predefined manner in data storage. Natural language processing (NLP) tools help understand unstructured data that exists in a written format.

In contrast, the meaning of structured data is data that follows predefined data models and is easy to analyze. Structured data examples would include alphabetically arranged names of customers and properly organized credit card numbers. After understanding the definition of unstructured data, let’s look at some examples.

Big Data includes huge volume, high velocity, and extensible variety of data. There are 3 types: Structured data, Semi-structured data, and Unstructured data.

Structured data –

Structured data is data whose elements are addressable for effective analysis. It has been organized into a formatted repository that is typically a database. It concerns all data which can be stored in database SQL in a table with rows and columns. They have relational keys and can easily be mapped into pre-designed fields. Today, those data are most processed in the development and simplest way to manage information. Example: Relational data.

Semi-Structured data –

Semi-structured data is information that does not reside in a relational database but that has some organizational properties that make it easier to analyze. With some processes, you can store them in the relation database (it could be very hard for some kind of semi-structured data), but Semi-structured exist to ease space. Example: XML data.

Unstructured data –

Unstructured data is a data which is not organized in a predefined manner or does not have a predefined data model, thus it is not a good fit for a mainstream relational database. So for Unstructured data, there are alternative platforms for storing and managing, it is increasingly prevalent in IT systems and is used by organizations in a variety of business intelligence and analytics applications. Example: Word, PDF, Text, Media logs.

Examples of Unstructured Data

Unstructured data can be anything that's not in a specific format. This can be a paragraph from a book with relevant information or a web page. An example of unstructured data could also be Log files that are not easy to separate. Social media comments and posts are also unstructured.
Here is an example of unstructured data from a log file.

38,P-R-38636-6-45,P-R-39105-1-11,P-R-38036-1-5,P-R-35697-1-13,P-R-35087-1-27,P-R-34341-1-9,P-R-33341-1-15,P-R-33110-1-29,P-R-31345-1-693,P-R-29076-1-6,P-R-28767-1-8,P-R-28540-2-8,P-R-28312-1-10,P-R-28069-1-27,P-R-28032-1-9,P-R-26562-1-12,P-R-26527-5-20,P-R-26164-1-11,P-R-25785-1-30,P-R-25095-9-70,P-R-23504-1-15,P-R-19719-5-41203
Wed Sep 23 2020 05:21:01 GMT+0500

Unstructured data is qualitative, not quantitative, so it is mostly categorical and characteristic in nature. For example, data from social media or websites can help predict future buying trends or determine the effectiveness of a marketing campaign. Another unstructured data analytics example is detecting patterns in scam emails and chat, which can be useful for enterprises in monitoring policy compliance. That's why businesses extract and store unstructured data in data warehouses (also called data lakes) for analysis.

1) Artificial Intelligence is about_____.
1.Playing a game on Computer
2.Making a machine Intelligent
3.Programming on Machine with your Own Intelligence
4.Putting your intelligence in Machine

2) Who is known as the -Father of AI"?
1.Fisher Ada
2.Alan Turing
3.John McCarthy
4.Allen Newell

3)Select the most appropriate situation for that a blind search can be used.

1.Real-life situation
2.Small Search Space
3.Complex game
4. All of the above

4. Ways to achieve AI in real-life are_____.
1.Machine Learning
2.Deep Learning
3.Both a & b
4.None of the abov

5. The main tasks of an AI agent are_____.
1.Input and Output
2.Moment and Humanly Actions
3.Perceiving, thinking, and acting on the environment
4.None of the above

Q.6 The best AI agent is one which_____
1.Needs user inputs for solving any problem
2.Can solve a problem on its own without any human intervention
3.Need a similar exemplary problem in its knowledge base
4.All of the above

Q.7 Which of the given element improve the performance of AI agent so that it can make better decisions?
1.Changing Element
2.Performance Element
3.Learning Element
4.None of the above

Q.8 How many types of Machine Learning are there?
1.1
2.2
3.3
4.4

Q.1 Select the type of data that can be Structured easily ?
Date Of Birth
Profile Photo
Screenshots
directions to the shops

Q.2 Select the unstructured data

Name
shipping time
Price of product
Product description

Q.3 Unstructured data can come from which of the following?
Facebook
Twitter
Presentations
All of these are corr

Q.4 What is structured data?

Structured data is a type of data that is huge in number and has many inaccurate values
Structured data is a type of data that is very less in number and can be stored in proper rows and columns
Structured data is a type of data that has inaccurate values but can be stored in rows and columns

Q.5 An example of structured data is ____.
age information
reason for a customer complaint
customer reviews
pictures of the good/serviceect.

Q.6 What is unstructured data?

Unstructured data is a type of data that is huge in number and has many inaccurate values
Unstructured data is a type of data that is very less in number and can be stored in proper rows and columns
Unstructured data is a type of data that has inaccurate values but can be stored in rows and columns

Q.6 What is semi-structured data?

Semi-structured data is a type of data that is huge in number and has many inaccurate values
Semi-structured data is a type of data that is very less in number and can be stored in proper rows and columns
Semi-structured data is a type of data that has inaccurate values but can be stored in rows and columns
Semi-structured data is a type of data which has contained the data of both types i.e., structured data and semi-structured data

Big Data Frameworks

. Hadoop

There are four components in the Hadoop ecosystem

HDFS: Stands for Hadoop Distributed File System, a file system that stores data on computers in a cluster. In simple words, it is a storage unit of Hadoop.

YARN: An acronym for Yet Another Resource Negotiator, a resource manager. It manages all computing resources in clusters and uses them to schedule user applications.

M**apReduce:** A programming model for processing data.

Hadoop Common: Hosts libraries and utilities and provide them to the above Hadoop components as required.

HDFS is the distributed file system in Hadoop for storing big data. MapReduce is the processing framework for processing vast data in the Hadoop cluster in a distributed manner. YARN is responsible for managing the resources amongst applications in the cluster.

. 2. Apache Spark

It is a multi-language analytics engine for big data processing. It works well with very massive datasets. Along with batch processing, it supports stream processing. It distributes data across multiple computers itself or with the help of other distributing tools.

Spark uses **in-memory caching**, which makes it a superfast framework than other cluster computing systems, such as Hadoop. The data processing in memory is just one-step — reading data into memory, performing operations, and writing the results back.

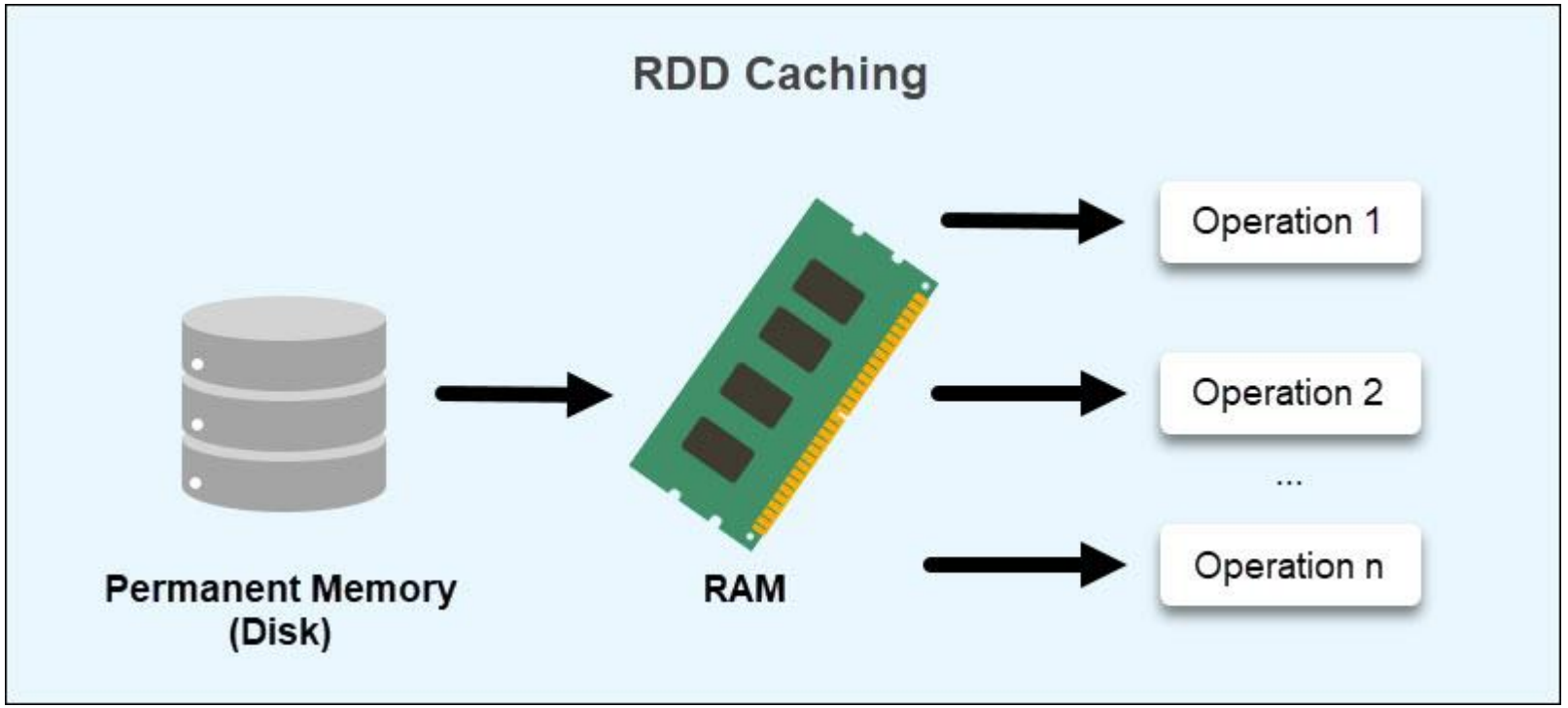
Resilient Distributed Dataset (RDD) forms the architectural basis for Spark. It is a read-only multi-set of data items spread across different machines in a cluster.

Based on	Hadoop	MapReduce
Definition	The Apache Hadoop is a software that allows all the distributed processing of large data sets across clusters of computers using simple programming	MapReduce is a programming model which is an implementation for processing and generating big data sets with distributed algorithm on a cluster.
Meaning	The name “Hadoop” was the named after Doug cutting’s son’s toy elephant. He named this project as “Hadoop” as it was easy to pronounce it.	The “MapReduce” name came into existence as per the functionality itself of mapping and reducing in key-value pairs.
Framework	Hadoop not only has storage framework which stores the data but creating name node’s and data node’s it also has other frameworks which include MapReduce itself.	MapReduce is a programming framework which uses a key, value mappings to sort/process the data
Invention	Hadoop was created by Doug Cutting and Mike Cafarella.	Mapreduce is invented by Google.
Features	• Hadoop is Open Source • Hadoop cluster is Highly Scalable	• Mapreduce provides Fault Tolerance • Mapreduce provides High Availability
Concept	The Apache Hadoop is an eco-system which provides an environment which is reliable, scalable and ready for distributed computing.	MapReduce is a submodule of this project which is a programming model and is used to process huge datasets which sits on HDFS (Hadoop distributed file system).
Language	Hadoop is a collection of all modules and hence may include other programming/scripting languages too	MapReduce is basically written in Java programming language

Why Do We Need RDDs in Spark?

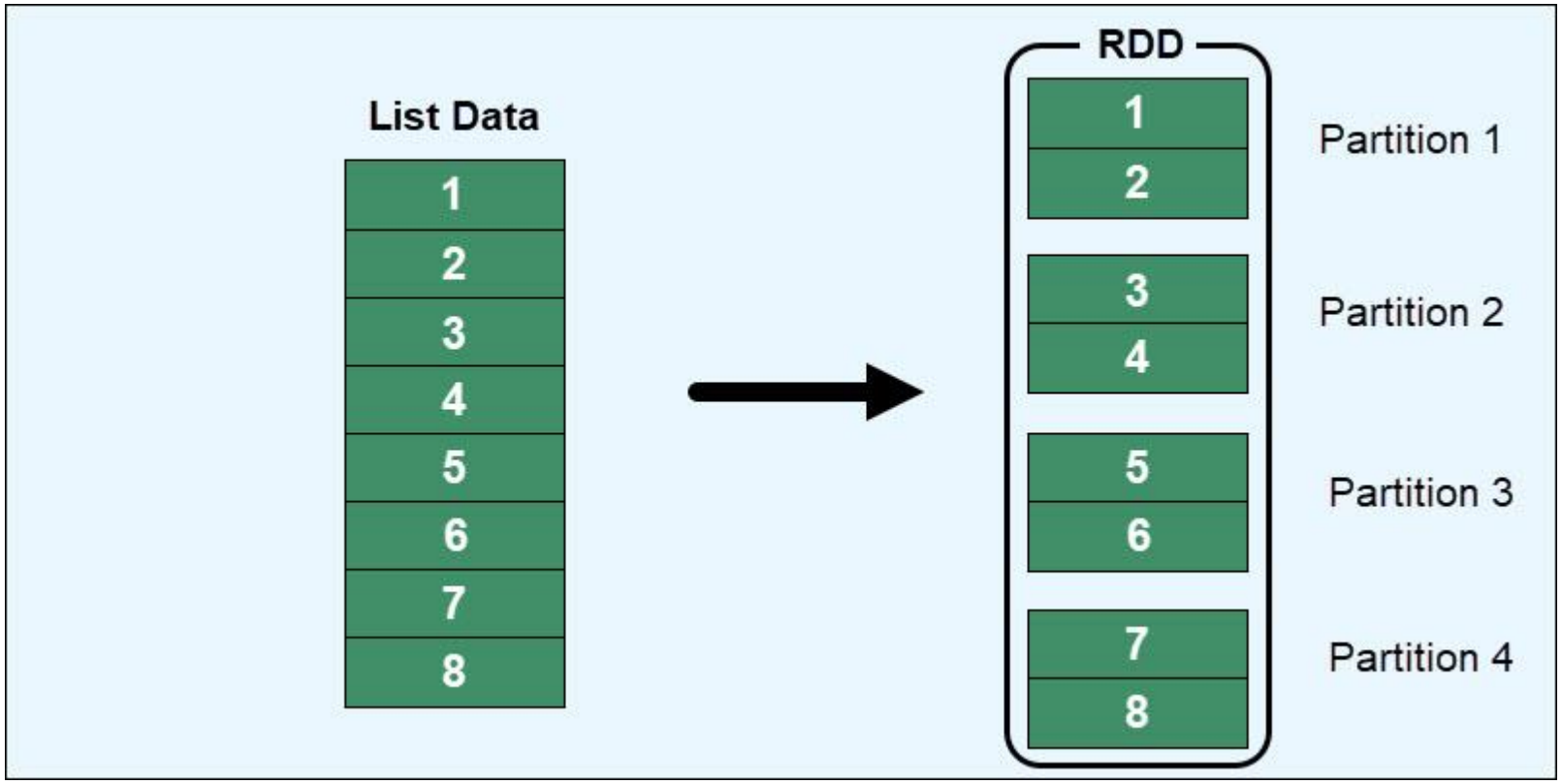
RDDs address MapReduce's shortcomings in data sharing. When reusing data for computations, MapReduce requires writing to external storage (HDFS, Cassandra, HBase, etc.). The read and write processes between jobs consume a significant amount of memory.

Furthermore, data sharing between tasks is slow due to replication, serialization, and increased disk usage.



RDDs aim to reduce the usage of external storage systems by leveraging **in-memory compute operation storage.** This approach improves data exchange speeds between tasks by 10 to 100 times. Speed is critical when working with large data volumes. Spark RDDs make it easier to train machine learning algorithms and handle large amounts of data for analytic

How Does RDD Store Data?



The disadvantages when working with Resilient Distributed Datasets include:

No schematic view of data. RDDs have a hard time dealing with structured data. A better option for handling structured data is through the DataFrames and Datasets APIs, which fully integrate with RDDs in Spark.

Garbage collection. Since RDDs are in-memory objects, they rely heavily on Java's memory management and serialization. This causes performance limitations as data grows.

Overflow issues. When RDDs run out of RAM, the information resides on a disk, requiring additional RAM and disk space to overcome overflow issues.

No automated optimization. An RDD does not have functions for automatic input optimization. While other Spark objects, such as DataFrames and Datasets, use the Catalyst optimizer, for RDDs, optimization happens manually.

Spark consists of the following core components

Spark Core: An execution engine and the heart of Spark, which forms the basis for all other components. It manages task dispatching, I/O operations, and task scheduling.

Spark SQL: Built on top of Spark core, Spark SQL performs distributed processing on data. It provides access to various data sources — HDFS, Hive, etc.

Spark Streaming: A library to process streaming data. It can stream gigabytes per second. It splits data into mini-batches and transforms them into RDDs.

MLlib: A machine learning containing different ML algorithms.

GraphX: A distributed graph-processing framework.

What is the difference between persist and cache table in Spark?

CACHE and PERSIST do the same job to help in retrieving intermediate data used for computation quickly by storing it in memory, while by caching we can store intermediate data used for calculation only in memory , persist additionally offers caching with more options/flexibility

3. Apache Hive

Hive is an open-source distributed data warehouse system built on top of Apache Hadoop. It supports reading, writing, and analyzing petabytes of data stored in distributed storage. We get an SQL-like interface called HiveQL to query data stored in databases and file systems that integrate with Hadoop.

Traditional databases can only process small to medium volumes of data. On the other hand, Hive leverages batch processing, like Hadoop, to process data quickly across a distributed database.

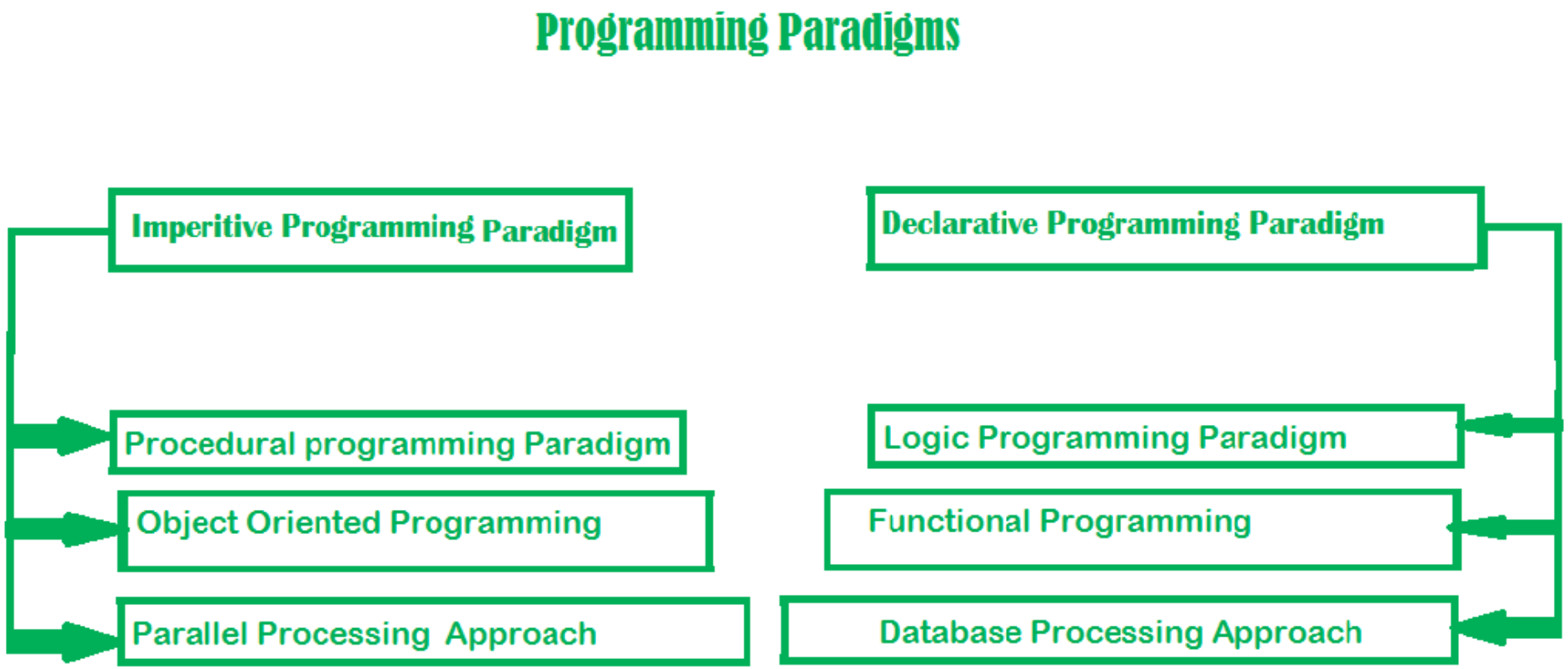
4.Apache Storm

Storm is a free and open-source real-time distributed big data processing system. It makes it easy to process unbounded streams of data. It processes data in a fault-tolerant and horizontally-scalable way. While simple to use, Storm is compatible with any programming language.

5. Apache Samza

It is a distributed stream processing framework, allowing users to build real-time applications that can process data.

Big Data Programming Paradigms



1. **Imperative programming paradigm:** It is one of the oldest programming paradigm. It features close relation to machine architecture. It is based on Von Neumann architecture. It works by changing the program state through assignment statements. It performs step by step task by changing state. The main focus is on how to achieve the goal. The paradigm consist of several statements and after execution of all the result is stored.

Advantages:
1. Very simple to implement
2. It contains loops, variables etc.

Disadvantage:
1.Complex problem cannot be solved
2.Less efficient and less productive
3.Parallel programming is not possible

Examples of Imperative programming paradigm:

C : developed by Dennis Ritchie and Ken Thompson
Fortran : developed by John Backus for IBM
Basic : developed by John G Kemeny and Thomas E Kurtz

2. Procedural programming paradigm –

This paradigm emphasizes on procedure in terms of under lying machine model. There is no difference in between procedural and imperative approach. It has the ability to reuse the code and it was boon at that time when it was in use because of its reusability.

Examples of Procedural programming paradigm:

C : developed by Dennis Ritchie and Ken Thompson
C++ : developed by Bjarne Stroustrup
Java : developed by James Gosling at Sun Microsystems
ColdFusion : developed by J J Allaire
Pascal : developed by Niklaus Wirth

3. Object oriented programming –

The program is written as a collection of classes and object which are meant for communication. The smallest and basic entity is object and all kind of computation is performed on the objects only. More emphasis is on data rather procedure. It can handle almost all kind of real life problems which are today in scenario.

Advantages:
•Data security
•Inheritance
•Code reusability
•Flexible and abstraction is also present

Examples of Object Oriented programming paradigm:

Simula : first OOP language
Java : developed by James Gosling at Sun Microsystems
C++ : developed by Bjarne Stroustrup
Objective-C : designed by Brad Cox
Visual Basic .NET : developed by Microsoft
Python : developed by Guido van Rossum
Ruby : developed by Yukihiro Matsumoto
Smalltalk : developed by Alan Kay, Dan Ingalls, Adele Goldberg

Parallel processing approach –

Parallel processing is the processing of program instructions by dividing them among multiple processors. A parallel processing system posses many numbers of processor with the objective of running a program in less time by dividing them. This approach seems to be like divide and conquer. Examples are NESL (one of the oldest one) and C/C++ also supports because of some library function.

2. Declarative programming paradigm:
It is divided as Logic, Functional, Database. In computer science the declarative programming is a style of building programs that expresses logic of computation without talking about its control flow. It often considers programs as theories of some logic.It may simplify writing parallel programs. The focus is on what needs to be done rather how it should be done basically emphasize on what code is actually doing. It just declares the result we want rather how it has be produced. This is the only difference between imperative (how to do) and declarative (what to do) programming paradigms. Getting into deeper we would see logic, functional and database.

Logic programming paradigms –

It can be termed as abstract model of computation. It would solve logical problems like puzzles, series etc. In logic programming we have a knowledge base which we know before and along with the question and knowledge base which is given to machine, it produces result. In normal programming languages, such concept of knowledge base is not available but while using the concept of artificial intelligence, machine learning we have some models like Perception model which is using the same mechanism.
In logical programming the main emphasize is on knowledge base and the problem. The execution of the program is very much like proof of mathematical statement,
e.g., Prolog

```
predicates
  sumoftwonumber(integer, integer).

clauses
  sumoftwonumber(0, 0).
  sumoftwonumber(N, R) :-
    N > 0,
    N1 is N - 1,
    sumoftwonumber(N1, R1),
    R is R1 + N.
```

Functional programming paradigms –

The functional programming paradigms has its roots in mathematics and it is language independent. The key principle of this paradigms is the execution of series of mathematical functions. The central model for the abstraction is the function which are meant for some specific computation and not the data structure. Data are loosely coupled to functions.The function hide their implementation. Function can be replaced with their values without changing the meaning of the program. Some of the languages like perl, javascript mostly uses this paradigm.

Examples of Functional programming paradigm:

JavaScript : developed by Brendan Eich
Haskell : developed by Lennart Augustsson, Dave Barton
Scala : developed by Martin Odersky
Erlang : developed by Joe Armstrong, Robert Virding
Lisp : developed by John Mccarthy
ML : developed by Robin Milner
Clojure : developed by Rich Hickey

Database/Data driven programming approach –

This programming methodology is based on data and its movement. Program statements are defined by data rather than hard-coding a series of steps. A database program is the heart of a business information system and provides file creation, data entry, update, query and reporting functions. There are several programming languages that are developed mostly for database application. For example SQL. It is applied to streams of structured data, for filtering, transforming, aggregating (such as computing statistics), or calling other programs. So it has its own wide application.

Big Data Programming Languages

Big Data Programming Languages

1. Python

Python is a general-purpose programming language that programmers use when integrating data analysis with web applications. Data scientists and technical analysts prefer using this open-source programming language because it offers them multiple data manipulation and plotting libraries, such as Pandas and Matplotlib. Data visualisation and machine learning become easy with Python, as it can support varied analyses and integrate with third-party packages. This high-level programming language may also work as a programming interface for an analytics system.

2. R

R is an open-source programming language that users utilise to work with graphics data visualisation and statistics. There is a wide range of graphical tools in R, along with open-source packages that help users visualise, model, manipulate and load data. Its robust environment allows a technical analyst to perform several types of data analyses. This programming language is highly flexible, as a user can run it on almost all operating systems.

3. Scala

Scala, also referred to as scalable language, is an efficient programming language for processing data quickly. It supports both functional programming and object-oriented programming (OOP), so it becomes easy for users to utilise languages based on these programming models. Several front-end developers prefer using Scala, as it combines compact syntax with impressive development tools. Fintech companies also utilise Scala to work with data architectures and cloud-based technologies. Scala has several features that a user can employ to write algorithms for machine learning and devise solutions for complex analytics.

4. Java

Programmers use Java to write production code that enables them to use big-data algorithms. Java for big data is helpful when programmers are implementing a theoretical model that they have created in Python. Big-data analysis is easy with Java, as it helps data scientists to process big data, manage higher prediction load and resize intricate ecosystems. Java also works as the base for many big-data tools, such as Spark, Storm or Mahout. Java is the foundation of Scala's Apache Spark library, so knowing how to work with it may also help users to write in Scala comfortably.

5. SQL

SQL stands for Structured Query Language, which is useful when working with complex datasets outside a relational-database environment. With SQL, a user can perform various operations, such as modifying tables and updating or removing records. It can help a data scientist to work with structured data. There are big-data platforms, such as Spark and Hadoop, that offer extension for querying using this domain-specific language. SQL is an effective tool with which data scientists can wrangle and prepare data, so they use the language when working with different big-data tools.

6. C++

When technical experts are working with complex machine-learning algorithms, they may often process data sets in terabytes and petabytes. To complete such tasks quickly, they may use C++, as this platform can process data in gigabytes in just a few seconds. Conducting predictive analytics in real time and keeping records consistent are some other benefits of using C++. A data scientist may use the programming language to code libraries and big-data frameworks. There are many deep-learning algorithms and neural networks that data scientists can write in C++.

7. Go

Go, also referred to as Golang, is an open-source programming language that helps developers build simple and efficient software tools. Its presence is usually evident in DevOps and web servers, but it is also a useful language for businesses that make data-driven decisions. A business can use Go to integrate computationally exhaustive algorithms with all its levels of organisational structure. With Go, pre-processing, transforming, analysing, modelling and validating become easy. This language allows users to write controllers for events that occur asynchronously.

8. Julia

Julia's performance is comparable to C++, which means it is fast, reliable and efficient. It is a programming language that offers robust statistical applicability and an interactive command line. C, R, Java and Python form the basis for many of its libraries. These libraries help data scientists to perform artificial intelligence development with ease. Doing high-level statistical work on this platform is unchallenging. It also outperforms some other languages when working with linear algebra, as it supports several machine-learning equations and matrixes.