General

○ Would you be comfortable working at night?

● HR Questions:

○ Your 3 weaknesses and strengths?

○ Expected Salary?

○ Are you OK with what we do.

○ What are your hobbies and stuff

○ Tell us something about yourself

○ Introduce yourself. Tell us about your educational and professional background (if any).

○ What is the perception of i2c among fastians.

○ What differentiates you from other candidates?

○ Biggest accomplishment?

○ Role in FYP?

○ Difference between smart work and hard work (what do you prefer? explain?)

○ Any kind of negative remarks about this company?

○ What makes you frustrated?

○ Did you participate in any co-curricular activity?

○ What are your passions?

○ You might get other job offers, so what would be the reasons to choose i2c over other companies?

○ Do you have any questions to ask from us?

○ Why is your CGPA low? (incase you have a low CGPA)

○ Do you have any plans for further studies?

○ You might also be asked something related to the essay you wrote in the HR assessment.

● Technical:

○ (Just basic CONCEPT of the following, not much detail.)

○ Basic intro of your FYP

○ What was your part in fyp

○ Basic intro of any other projects you have done

○ Tell some git commands

- **git config**
- **git init**
- **git clone**
- **git add**
- **git commit**
- **git diff**
- **git reset**
- **git status**
- **git rm**
- **git log**
- **git show**
- **git tag**
- **git branch**
- **git checkout**
- **git merge**
- **git remote**
- **git push**
- **git pull**
- **git stash**

○ What is difference b/w encryption and compression

Even though both data compression and encryption are methods that transform data in to a different format, the golas tried to achieve by them are different. Data compression is done with the intension of decreasing the size of data, while encryption is done to keep the data secret from third parties. Encrypted data cannot be decrypted easily. It requires the possession of a special piece of information called a key. Uncompressing compressed data do not require such special knowledge (such as a key), but it might require some special hardware depending on the data type.

○ Tell some encryption algos

- Triple DES
- RSA
- BlowFish
- TwoFish
- AES

○ How would you encrypt any given string?

Given a string **s**, the task is to encrypt the string in the following way:

1. If the frequency of current character is even, then increment current character by x.
2. If the frequency of current character is odd, then decrement current character by x.

○ What is backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.

Items are ordered by priority

○ How long is a Sprint

In Agile product development, a sprint is a set period of time during which specific work has to be completed and...

Sprint is 2-4 weeks long (agile sprint is no more than one month long)

○ DETAILED explanation of FYP

○ Applications of FYP

○ Name some non-functional requirements of your FYP (+ name some general non-functional requirements).

○ Why do we use ftp protocol

File transfer protocol is protocol to transfer files on network from one machine to another

○ What is serialization and deserialization?

Serialization in Java is a mechanism of *writing the state of an object into a byte-stream*.

The reverse operation of serialization is called *deserialization* where byte-stream is converted into an object

Java provides a mechanism, called object serialization where an object can be represented as a sequence of bytes that includes the object's data as well as information about the object's type and the types of data stored in the object.

After a serialized object has been written into a file, it can be read from the file and deserialized that is, the type information and bytes that represent the object and its data can be used to recreate the object in memory.

○ Regex and How to use regex on a particular initial?

Regular Expressions (also called Regex or Regexp) is a pattern in which the rules for matching text are written in form of metacharacters, quantifiers or plain text. They are strings in which "what to match" is defined or written. Regex is used for finding patterns or replacing the matched patterns. It is

○ Write recursive factorial function?

```
int factorial(int n)
{
    if(n > 1)
        return n * factorial(n - 1);
    else
        return 1;
}
```

○ Difference between array and vectors?

Vector is synchronized and array is not synchronized.

The Vector class implements a growable array of objects. Vectors basically fall in legacy classes but now it is fully compatible with collections.

- Vector implements a dynamic array that means it can grow or shrink as required. Like an array, it contains components that can be accessed using an integer index
- They are very similar to ArrayList but Vector is synchronised and have some legacy method which collection framework does not contain.
- It extends **AbstractList** and implements **List** interfaces.

Vector implements a dynamic array. It is similar to ArrayList, but with two differences −

- Vector is synchronized.
- Vector contains many legacy methods that are not part of the collections framework.

Vector proves to be very useful if you don't know the size of the array in advance or you just need one that can change sizes over the lifetime of a program.

○ What are collections?

The **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects.

Java Collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion.

Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet).

- Java Collection Framework
- Hierarchy of Collection Framework
- Collection interface
- Iterator interface

○ What is static and final?

The main difference between a static and final keyword is that **static** is keyword is used to define the class member that can be used independently of any object of that class. **Final** keyword is used to declare, a constant variable, a method which can not be overridden and a class that can not be inherited.

○ What is garbage collection? How would you implement it?

In computer science, **garbage collection** (**GC**) is a form of automatic memory management. The *garbage collector*, or just *collector*, attempts to reclaim *garbage*, or memory occupied by objects that are no longer in use by the program.

The technique that accomplishes this is known as **Garbage Collection**. Programs that do not de-allocate memory can eventually crash when there is no memory left in the system to allocate. These programs are said to have *memory leaks.*
When there are no references to an object, it is assumed to be no longer needed, and the memory, occupied by the object can be reclaimed. There is no explicit need to destroy an object as Java handles the de-allocation automatically.

Main objective of Garbage Collector is to free heap memory by destroying **unreachable objects**.

1. **Unreachable objects :** An object is said to be unreachable iff it doesn't contain any reference to it. Also note that objects which are part of island of isolation are also unreachable.

1) If you want to make your object eligible for Garbage Collection, assign its reference variable to null.

○ What is CNN?

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

○ How would you make an artificially intelligent machine?
○ What are access specifiers?

Access specifiers define how the members (attributes and methods) of a class can be accessed.
Jis sy ham class ky variables or methods ki access ko control karty han

○ Which protocol should be used while making a chat system ?
XMPP(extensible message presence protocol)
Extensible Messaging and Presence Protocol (abbreviated as XMPP) is an open source and extensible protocol that was originally referred as Jabber. XMPP protocol supports the transmission of current information such as data. As a messaging protocol, it can only be applied effectively by moving through an appropriate transport binding such as TCP/IP, HTTP, or WebSocket. Some of the XMPP applications include Gtalk and Whatsapp.
○ How can you create a heap overflow ?

Heap Overflow:

Heap is a region of process's memory which is used to store dynamic variables. These variables are allocated using malloc() and calloc() functions and resize using realloc() function, which are inbuilt functions of C. These variables can be accessed globally and once we allocate memory on heap it is our responsibility to free that memory space after use. There are two situations which can result in heap overflow:

1. If we continuously allocate memory and we do not free that memory space after use it may result in memory leakage – memory is still being used but not available for other processes.
2. if we dynamically allocate large number of variables

Stack overflow

Stack is a special region of our process's memory which is used to store local variables used inside the function, parameters passed through a function and their return addresses. Whenever a new local variable is declared it is pushed onto the stack. All the

variables associated with a function are deleted and memory they use is freed up, after the function finishes running. The user does not have any need to free up stack space manually. Stack is Last-In-First-Out data structure.

1.If we declare large number of local variables or declare an array or matrix or any higher dimensional array of large size can result in overflow of stack.

2. If function recursively call itself infinite times then the stack is unable to store large number of local variables used by every function call and will result in overflow of stack.

- Database
  - Normal Forms.
  - Anomalies
  - DDL, DML
  - Views,Triggers
  - Write a query to delete a table

    Drop table tablename
  - How will you create a table with images?

We will first create a table named myimages in SQL Server:

```
CREATE TABLE myimages(id int, img varbinary(max))
```

This table will have an integer (int) id and the image column named img. The data type that we are going to use to store images is the varbinary(max).

```
INSERT INTO BLOBTest
        (BLOBName, BLOBData)
        SELECT 'First test file',
                BulkColumn FROM OPENROWSET(
                        Bulk 'C:\temp\nextup.jpg', SINGLE_BLOB) AS BLOB
```

  - When do we use varchar

CHAR

1. Used to store character string value of **fixed length**.
2. The maximum no. of characters the data type can hold is **255 characters**.
3. It's **50% faster** than VARCHAR.
4. Uses **static memory allocation**.

VARCHAR

1. Used to store **variable length** alphanumeric data.
2. The maximum this data type can hold is up to
   - Pre-MySQL 5.0.3: **255 characters**.
   - Post-MySQL 5.0.3: **65,535 characters** shared for the row.
3. It's **slower** than CHAR.
4. Uses **dynamic memory allocation**.

  - What is noSQL

NoSQL databases are databases that store data in a format other than relational tables.

NoSQL databases allow developers to store huge amounts of unstructured data, giving them a lot of flexibility.

  - What are transactions
  - What is rollback
  - How would you implement a rollback function if you are creating a DBMS?
  - What is SQL injection?
  - Truncate vs Delete vs DROP
  - Write a query to select average salary department wise

○ Delete a row
○ How are views helpful/Advantage of views in business layer?
○ Why is TRUNCATE a DDL?

TRUNCATE actually drops & re-creates the table, and resets the table's metadata

○ How to use a BLOB?

BLOB :

`BLOB` (*Binary Large Object*) is a large object data type in the database system. `BLOB` could store a large chunk of data, document types and even media files like audio or video files. `BLOB` fields allocate space only whenever the content in the field is utilized. `BLOB` allocates spaces in Giga Bytes.

USAGE OF BLOB :

You can write a binary large object (`BLOB`) to a database as either binary or character data, depending on the type of field at your data source. To write a `BLOB` value to your database, issue the appropriate `INSERT or UPDATE` statement and pass the `BLOB` value as an input parameter. If your `BLOB` is stored as text, such as a SQL Server text field, you can pass the `BLOB` as a string parameter. If the `BLOB` is stored in binary format, such as a SQL Server image field, you can pass an array of type byte as a binary parameter.

You can also store large files as disk files and URL's in columns but this can result in loss of track and movement of files along with the database.

○ What is ACID?
○ What is projection?

Selection selects rows based on condition(s) specified
Projection selects all rows for the specified columns

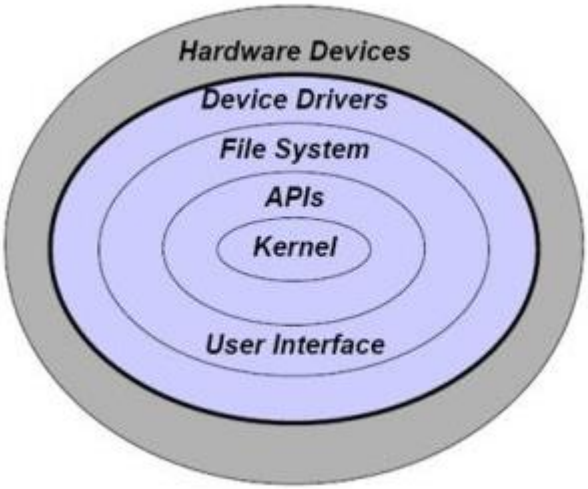
○ How to delete a column from a table?

```
ALTER TABLE Customers
DROP COLUMN ContactName;
```

● O.S:
○ Type of File Systems (just names)

controls how data is stored and retrieved(ways of organizing and storing files on hard drive.
windows file sytems are FAT(file allocation table) and NTFS(new technology file system)
Ubuntu's is ext
macOS are HFS+ and APFS
Unix is UFS and FFS

○ components of List operating systems



○ List data structures used in OS.
○ Where is a heap used in OS.
○ How does a file system run/work
○ What are some commands of assembly language?
○ What is fragmentation?

https://www.includehelp.com/operating-systems/fragmentation.aspx
https://www.tutorialspoint.com/difference-between-internal-fragmentation-and-external-fragmentation

○ List some terminal commands
● C.N:
○ What are ports?

In computer networking, a **port** is a communication endpoint
The most common protocols that use port numbers are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).
Higher-numbered ports are available for general use by applications and are known as ephemeral ports.
0-65535

○ List network protocol

https://www.w3schools.in/types-of-network-protocols-and-their-uses/

○ Which protocol would you use if you wanted to create Twitter? (ans: HTTP, not TCP)
https
○ How to prevent a DDos attack ?

A DDoS attack happens when a hacker sends a flood of traffic to a network or server in order to overwhelm(overload) the system and disrupt its ability to operate.

If you know what to expect when your traffic hits its upper limit, you can put rate limiting into place. **That means that the server will only accept as many requests as it can handle.**

Another technique to prevent attacks is to hire a DDoS mitigation service. Their way of preventing attacks is to route all incoming traffic through a filter, so that only genuine traffic hits your website or application.

Increase Internet Bandwidth and Server Capacity

Using web application firewalls is a great way to protect larger enterprise-level applications. A firewall can detect and prevent DDoS attacks by monitoring unusual spikes in traffic and blocking them.

A content distribution network, or CDN, can balance out website traffic by spreading it across different servers located around the globe.

- C.A:
  - Fragmentation.
  - Virtual Memory
  - What data structure would you use to implement Main Memory?
  - Caches are different from Main Memory, so what data structure would you use to implement a cache?
- OOP:
  - Static Variable
  - Singleton Class
  - What is open/closed principle

Software entities should be open for extension, but closed for modification.

> *"Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification."*

> The general idea of this principle is great. It tells you to write your code so that you will be able to add new functionality without changing the existing code. That prevents situations in which a change to one of your classes also requires you to adapt all depending classes. Unfortunately, Bertrand Mayer proposes to use inheritance to achieve this goal:

> *"A class is closed, since it may be compiled, stored in a library, baselined, and used by client classes. But it is also open, since any new class may use it as parent, adding new features. When a descendant class is defined, there is no need to change the original or to disturb its clients."*

  - What is cohesion

Coupling and **cohesion** are terms which occur together very frequently. **Coupling** refers to the interdependencies between modules, while **cohesion** describes how related the functions within a single module are.

  - What is single responsibility principle

- Single Responsibility Principle
- Open/Closed Principle
- Liskov Substitution Principle
- Interface Segregation Principle
- Dependency Inversion

In programming, the Single Responsibility Principle states that every module or class should have responsibility over a single part of the functionality provided by the software.

- Composition vs inheritance

Classes and objects created through inheritance are *tightly coupled* because changing the parent or superclass in an inheritance relationship risks breaking your code. Classes and objects created through composition are *loosely coupled*, meaning that you can more easily change the component parts without breaking your code.

- What is coupling?
- How can you increase coupling? How would you increase coupling in a SINGLE class?
- Is coupling high in composition?
- How can you reduce coupling?
- What are immutable strings in Java.

An immutable class is simply a class whose instances cannot be modified

An important point to note here is that, while the `String` object is immutable, **its reference variable is not.** So that's why, in the above example, the reference was made to refer to a newly formed `String` object.

Because java uses the concept of string literal.Suppose there are 5 reference variables,all referes to one object "sachin".If one reference variable changes the value of the object, it will be affected to all the reference variables. That is why string objects are immutable in java.

- Data Structures:
  - Basic definition of stack and queue.

Stack A stack is a linear data structure in which elements can be inserted and deleted only from one side of the list, called the **top**. A stack follows the **LIFO** (Last In First Out) principle, i.e., the element inserted at the last is the first element to come out. The insertion of an element into stack is called **push** operation, and deletion of an element from the stack is called **pop** operation.

A queue is a linear data structure in which elements can be inserted only from one side of the list called **rear**, and the elements can be deleted only from the other side called the **front**. The queue data structure follows the **FIFO** (First In First Out) principle, i.e. the element inserted at first in the list, is the first element to be removed from the list. The insertion of an element in a queue is called an **enqueue** operation and the deletion of an element is called a **dequeue** operation. In queue we always maintain two pointers, one pointing to the element which was inserted at the first and still present in the list with the **front** pointer and the second pointer pointing to the element inserted at the last with the **rear** pointer.

- Name some searching algorithms.
  - Linear
  - Binary

- Explain a code to delete second last element of a linked list
- What is the difference between a set and a list/array?

List is a type of ordered collection that maintains the elements **in** insertion order while **Set** is a type of unordered collection so elements are not maintained any order. **List** allows duplicates while **Set** doesn't allow duplicate elements

he biggest **difference between an Array** & **Set** is that **Arrays** can have duplicate values whereas **Sets** cannot. The other big **difference** is that data **in an array** is ordered by index whereas **Sets** use keys & the elements are iterable **in the** order of insertion.Oct 8, 2018

- Java:
  - What is the role of jdbc/odbc?

    JDBC Driver is a software component that enables java application to interact with the database.

ODBC Stands for Open Database Connectivity.	JDBC Stands for java database connectivity.
Introduced by Microsoft in 1992.	Introduced by SUN Micro Systems in 1997.
We can use ODBC for any language like C,C++,Java etc.	We can use JDBC only for Java languages.
We can choose ODBC only windows platform.	We can Use JDBC in any platform.
Mostly ODBC Driver developed in native languages like C,C++.	JDBC Stands for java database connectivity.
For Java applications it is not recommended to use ODBC because performance will be down due to internal conversion and applications will become platform Dependent.	For Java application it is highly recommended to use JDBC because there we no performance & platform dependent problem.
ODBC is procedural.	JDBC is object oriented.

  - What is the difference between immutable and mutable strings? (Strings are always immutable!)
  - How can you make a String mutable?

With Mutable string, we can change the contents of an existing object which does not result in the creation of a new object.
As already covered, the mutable string in java can be created using StringBuffer and StringBuilder classes.

No.	StringBuffer	StringBuilder
1)	StringBuffer is *synchronized* i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously.	StringBuilder is *non-synchronized* i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously.
2)	StringBuffer is *less efficient* than StringBuilder.	StringBuilder is *more efficient* than StringBuffer.

  - Null pointer exception

    NullPointerException is thrown when program attempts to use an object reference that has the null value.
    To avoid the NullPointerException, we must ensure that all the objects are initialized properly, before you use them

- Invoking a method from a null object.
- Accessing or modifying a null object's field.
- Taking the length of null, as if it were an array.
- Accessing or modifying the slots of null object, as if it were an array.
- Throwing null, as if it were a Throwable value.
- When you try to synchronize over a null object.

○ What are packages in Java?

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.

2) Java package provides access protection.

3) Java package removes naming collision.

○ What is final in java why we make a final method ?
So that it cannot be overriden

● Software Testing:
○ Describe the types of ST
https://www.softwaretestinghelp.com/types-of-software-testing/

○ What is unit testing

UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class.

○ What is assert statements (in junit)
An assert statement is used to declare an expected boolean condition in a program. If the program is running with assertions enabled, then the condition is checked at runtime. If the condition is false, the Java runtime system throws an `AssertionError`.

○ How did you perform white box and black box testing (if you have done it)

Black Box Testing is a software **testing** method in which the internal structure/ design/ implementation **of the** item being tested is not known to the **tester**. **White Box Testing** is a software **testing** method in which the internal structure/ design/ implementation **of the** item being tested is known to the **tester**.

● Assembly:
○ What does shift left operation do mathematically? (Ans: multiplication by 2)
● Software Engineering:
○ What is SOLID Principle?

- Software reengineering?

Software Reengineering is the process of updating **software** without affecting its functionality. This process may be done by developing additional features on the **software** and adding functionalities that may or may not be required but considered to make the **software** experience better and more efficient.

UML diagrams

Unified modelling language

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on **diagrammatic representations** of software components.

- Structure Diagrams
  - Class Diagram
  - Component Diagram
  - Deployment Diagram
  - Object Diagram
  - Package Diagram
  - Profile Diagram
  - Composite Structure Diagram

- Behavioral Diagrams
  - Use Case Diagram
  - Activity Diagram
  - State Machine Diagram
  - Sequence Diagram
  - Communication Diagram
  - Interaction Overview Diagram
  - Timing Diagram

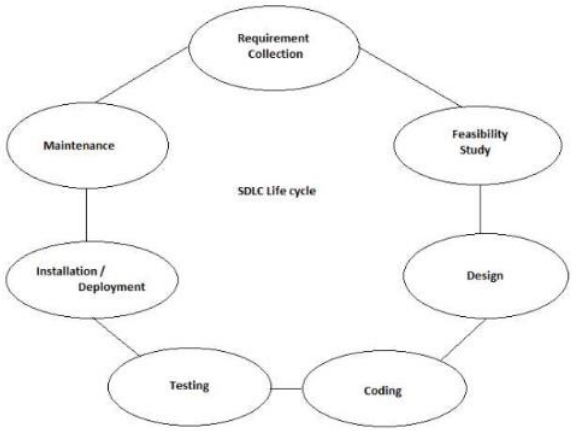
Exception?

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions.

Error: An Error indicates serious problem that a reasonable application should not try to catch.

Exception: Exception indicates conditions that a reasonable application might try to catch.

SDLC is a step by step procedure or systematic approach to develop software and it is followed within a software organization. It consists of various phases which describe how to design, develop, enhance and maintain particular software.



AGILE???

AGILE methodology is a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent unlike the Waterfall model

> Agile is a term used to describe approaches to software development emphasizing incremental delivery, team collaboration, continual planning, and continual learning, instead of trying to deliver it all at once near the end.

Waterfall model??

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Prototype Model

This model is used when the customers do not know the exact project requirements beforehand.

This prototype is developed based on the currently known requirements. Prototype model is a **software development model**. By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.

Session Management

HTTP protocol and Web Servers are stateless, what it means is that for web server every request is a new request to process and they can't identify if it's coming from client that has been sending request previously.

Session is a conversional state between client and server and it can consists of multiple request and response between client and server. Since HTTP and Web Server both are stateless, the only way to maintain a session is when some unique information about the session (session id) is passed between server and client in every request and response.

JSON vs XML

JSON	XML
JSON object has a type	XML data is typeless
JSON types: string, number, array, Boolean	All XML data should be string
Data is readily accessible as JSON objects	XML data needs to be parsed.
JSON is supported by most browsers.	Cross-browser XML parsing can be tricky
JSON has no display capabilities.	XML offers the capability to display data because it is a markup language.
JSON supports only text and number data type.	XML support various data types such as number, text, images, charts, graphs, etc. It also provides options for transferring the structure or format of the data with actual data.
Retrieving value is easy	Retrieving value is difficult
Supported by many Ajax toolkit	Not fully supported by Ajax toolkit
A fully automated way of deserializing/serializing JavaScript.	Developers have to write JavaScript code to serialize/de-serialize from XML
Native support for object.	The object has to be express by conventions - mostly missed use of attributes and elements.
It supports only UTF-8 encoding.	It supports various encoding.
It doesn't support comments.	It supports comments.
JSON files are easy to read as compared to XML.	XML documents are relatively more difficult to read and interpret.
It does not provide any support for namespaces.	It supports namespaces.
It is less secured.	It is more secure than JSON.

Verification vs Validation

Verification is a static practice of verifying documents, design, code and program. It includes all the activities associated with producing high quality software: inspection, design analysis and specification analysis. It is a relatively objective process.

Verification will help to determine whether the software is of high quality, but it will not ensure that the system is useful. Verification is concerned with whether the system is well-engineered and error-free.

***Definition**: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.*

Validation is the process of evaluating the final product to check whether the software meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product.

Verification	*Validation*
1. **Verification** is a static practice of verifying documents, design, code and program.	1. **Validation** is a dynamic mechanism of validating and testing the actual product.
2. It does not involve executing the code.	2. It always involves executing the code.
3. It is human based checking of documents and files.	*3.* It is computer based execution of program.
4. **Verification** uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.	*4.* **Validation** uses methods like black box (functional) testing, gray box testing, and white box (structural) testing etc.
5. **Verification** is to check whether the software conforms to specifications.	5. **Validation** is to check whether software meets the customer expectations and requirements.
6. It can catch errors that validation cannot catch. It is low level exercise.	*6. It can catch errors that verification cannot catch. It is High Level Exercise.*
7. Target is requirements specification, application and software architecture, high level, complete design, and database design etc.	*7.* Target is actual product-a unit, a module, a bent of integrated modules, and effective final product.
8. **Verification** is done by QA team to ensure that the software is as per the specifications in the SRS document.	*8.* ***Validation** is carried out with the involvement of testing team.*
9. It generally comes first-done before validation.	*9.* It generally follows after **verification**.

Differences between a Stored Procedure and a Trigger

1. We can execute a stored procedure whenever we want with the help of the `exec` command, but a trigger can only be executed whenever an event (insert, delete, and update) is fired on the table on which the trigger is defined.
2. We can call a stored procedure from inside another stored procedure but we can't directly call another trigger within a trigger. We can only achieve nesting of triggers in which the action (insert, delete, and update) defined within a trigger can initiate execution of another trigger defined on the same table or a different table.
3. Stored procedures can be scheduled through a job to execute on a predefined time, but we can't schedule a trigger.
4. Stored procedure can take input parameters, but we can't pass parameters as input to a trigger.
5. Stored procedures can return values but a trigger cannot return a value.
6. We can use Print commands inside a stored procedure for debugging purposes but we can't use print commands inside a trigger.
7. We can use transaction statements like begin transaction, commit transaction, and rollback inside a stored procedure but we can't use transaction statements inside a trigger.
8. We can call a stored procedure from the front end (.asp files, .aspx files, .ascx files, etc.) but we can't call a trigger from these files.
9. Stored procedures are used for performing tasks. Stored procedures are normally used for performing user specified tasks. They can have parameters and return multiple results sets.
10. The Triggers for auditing work: Triggers normally are used for auditing work. They can be used to trace the activities of table events.

Stored Procedure Syntax

```
CREATE PROCEDURE procedure_name
AS
sql_statement
GO;
```

Execute a Stored Procedure

```
EXEC procedure_name;
```

```
CREATE TRIGGER new_employee
AFTER INSERT ON hr_table
BEGIN
    INSERT INTO benefits_table ('name','street','workid')
    VALUES ('john', '123 Any Street', '1022')
END
```

Following are the main differences between functions and procedures:

Functions	Procedures
A function has a return type and returns a value.	A procedure does not have a return type. But it returns values using the OUT parameters.
You cannot use a function with Data Manipulation queries. Only Select queries are allowed in functions.	You can use DML queries such as insert, update, select etc... with procedures.
A function does not allow output parameters	A procedure allows both input and output parameters.
You cannot manage transactions inside a function.	You can manage transactions inside a function.
You cannot call stored procedures from a function	You can call a function from a stored procedure.
You can call a function using a select statement.	You cannot call a procedure using select statements.

Client-server arhitecture

Client/server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or internet connection. This system shares computing resources.

MVC

The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

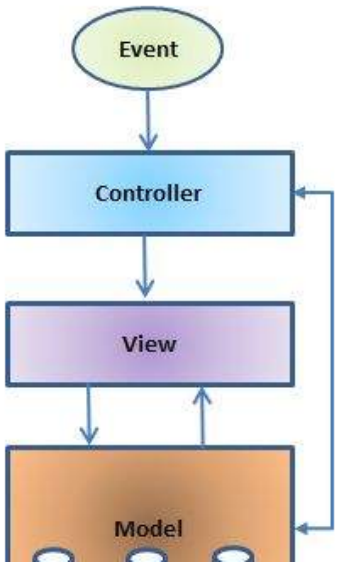
MVC Components

Following are the components of MVC −

Model **V**iew **C**ontroller or **MVC** as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts −

- **Model** − The lowest level of the pattern which is responsible for maintaining data.
- **View** − This is responsible for displaying all or a portion of the data to the user.
- **Controller** − Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.



Advantages of MVC architecture:

- Development of the application becomes fast.
- Easy for multiple developers to collaborate and work together.
- Easier to Update the application.
- Easier to Debug as we have multiple levels properly written in the application.

Disadvantages of MVC architecture:

- It is hard to understand the MVC architecture.
- Must have strict rules on methods.

There is not much in the disadvantages part of the architecture. And the disadvantages are not so huge and are very easy to ignore in comparison with all the benefits we get.

A package in Java is used to group related classes. Think of it as **a folder in a file directory**. We use packages to avoid name conflicts, and to write a better maintainable code. Packages are divided into two categories:

- Built-in Packages (packages from the Java API)
- User-defined Packages (create your own packages)

Another **real life example of Abstraction** is ATM Machine; All are performing operations on the ATM machine like cash withdrawal, money transfer, retrieve mini-statement…etc. but we can't know internal details about ATM. Note: Data **abstraction** can be used to provide security for the data from the unauthorized methods.

Real life example of polymorphism: A person at the same time can have different characteristic. Like a man at the same time is a father, a husband, an employee. So the same person posses different behaviour in different situations. This is called polymorphism.

- Here automatic cola vending machine is a class. It contains both data i.e. Cola can and operations i.e. service mechanism and they are wrapped/integrated under a single unit Cola Vending Machine. This is called **Encapsulation.**
- You need not know how the machine is working. This is called **Abstraction.**
- You can interact with cola can only through service mechanism. You cannot access the details about internal data like how much cans it contains, mechanism etc. This is **Data Hiding.**
- You cannot pick the can directly. You request for cola through proper instructions and request mechanism (i.e. by paying amount and filling request) and get that cola only through specified channel. This is **message passing.**

The term **Web service** (**WS**) is either:

- a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web, or
- a server running on a computer device, listening for requests at a particular port over a network, serving web documents (HTML, JSON, XML, images), and creating[*clarification needed*] web applications services, which serve in solving specific domain problems over the Web (WWW, Internet, HTTP)

application program interface

An **application program interface** (API) is a set of routines, protocols, and tools for building software applications. Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components.

2) Method Overloading: changing data type of arguments

In this example, we have created two methods that differs in data type. The first add method receives two integer arguments and second add method receives two double arguments.

```
class Adder{
static int add(int a, int b){return a+b;}
static double add(double a, double b){return a+b;}
}
class TestOverloading2{
public static void main(String[] args){
System.out.println(Adder.add(11,11));
System.out.println(Adder.add(12.3,12.6));
}}
```

```
class Bank{
int getRateOfInterest(){return 0;}
}
//Creating child classes.
class SBI extends Bank{
int getRateOfInterest(){return 8;}
}

class ICICI extends Bank{
int getRateOfInterest(){return 7;}
}
class AXIS extends Bank{
int getRateOfInterest(){return 9;}
}
//Test class to create objects and call the methods
class Test2{
public static void main(String args[]){
SBI s=new SBI();
ICICI i=new ICICI();
AXIS a=new AXIS();
System.out.println("SBI Rate of Interest: "+s.getRateOfInterest());
System.out.println("ICICI Rate of Interest: "+i.getRateOfInterest());
System.out.println("AXIS Rate of Interest: "+a.getRateOfInterest());
}
}
```

1. **POST** – This would be used to create a new employee using the RESTful web service
2. **GET** - This would be used to get a list of all employee using the RESTful web service
3. **PUT** - This would be used to update all employee using the RESTful web service
4. **DELETE** - This would be used to delete all employee using the RESTful web service

Sockets are part of transport layer, which provides logical communication to applications. This means that from application's point of view both hosts are directly connected to each other, even though there are numerous routers and/or switches between them. Thus a socket is not a connection itself, it's the end point of the connection. Transport layer protocols are implemented only on hosts, and not on intermediate routers.

Ports provide means of internal addressing to a machine. The primary purpose it to allow multiple processes to send and receive data over the network without interfering with other processes (their data). All sockets are provided with a port number. When a segment arrives to a host, the transport layer examines the destination port number of the segment. It then forwards the segment to the corresponding socket. This job of delivering the data in a transport layer segment to the correct socket is called **de-multiplexing**. The segment's data is then forwarded to the process attached to the socket.

Socket: "IP address and Port " together is called "Socket". It is used by another computer to send data to one particular computer's particular software.

Difference between Inheritance & Polymorphism

Basic	Inheritance is creating a new class using the properties of the already existing class.	Polymorphism is basically a common interface for multiple form.
Implementation	Inheritance is basically implemented on classes.	Polymorphism is basically implemented on function/methods.
Use	To support the concept of reusability in OOP and reduces the length of code.	Allows object to decide which form of the function to be invoked when, at compile time(overloading) as well as run time(overriding).
Forms	Inheritance may be a single inheritance, multiple inheritance, multilevel inheritance, hierarchical inheritance and hybrid inheritance.	Polymorphism may be a compile time polymorphism (overloading) or run-time polymorphism (overriding).
Example	The class 'table' can inherit the feature of the class 'furniture', as a 'table' is a 'furniture'.	The class 'study_table' can also have function 'set_color()' and a class 'Dining_table' can also have function 'set_color()' so, which form of the set_color() function to invoke can be decided at both, compile time and run time.

What is JavaScript?

JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

JavaScript was initially created to "make web pages alive".

The programs in this language are called *scripts*. They can be written right in a web page's HTML and run automatically as the page loads.

Scripts are provided and executed as plain text. They don't need special preparation or compilation to run.

In this aspect, JavaScript is very different from another language called Java.

For instance, in-browser JavaScript is able to:

- Add new HTML to the page, change the existing content, modify styles.
- React to user actions, run on mouse clicks, pointer movements, key presses.
- Send requests over the network to remote servers, download and upload files (so-called AJAX and COMET technologies).
- Get and set cookies, ask questions to the visitor, show messages.
- Remember the data on the client-side ("local storage").

What Is a Front-End Developer?

> Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The front end of a website is the part that users interact with. Everything that you see when you're navigating around the Internet, from fonts and colors to dropdown menus and sliders, is a combo of HTML, CSS, and JavaScript being controlled by your computer's browser.

HTML, CSS, & JavaScript:

A front-end developer architects and develops websites and applications using web technologies (i.e., HTML, CSS, DOM, and JavaScript), which run on the Open Web Platform or act as compilation input for non-web platform environments (i.e., React Native).

What is a Back-End Developer?

So what makes the front end of a website possible? Where is all that data stored? This is where the back end comes in. The back end of a website consists of a server, an application, and a database. A back-end developer builds and maintains the technology that powers those components which, together, enable the user-facing side of the website to even exist in the first place.

SKILLS AND TOOLS

In order to make the server, application, and database communicate with each other, back-end devs use server-side languages like PHP, Ruby, Python, Java, and .Net to build an application, and tools like MySQL, Oracle, and SQL Server to find, save, or change data and serve it back to the user in front-end code. Job openings for back-end developers often also call for experience with PHP frameworks like Zend, Symfony, and CakePHP; experience with version control software like SVN, CVS, or Git; and experience with Linux as a development and deployment system.

What is HTML

First developed by Tim Berners-Lee in 1990, **HTML** is short for **Hypertext Markup Language**. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

What is CSS? And How Does It Relate to HMTL?

CSS stands for Cascading Style Sheets with an emphasis placed on "Style." While HTML is used to structure a web document (defining things like headlines and paragraphs, and allowing you to embed images, video, and other media), CSS comes through and specifies your document's style—page layouts, colors, and fonts are all determined with CSS. Think of HTML as the foundation (every house has one), and CSS as the aesthetic choices (there's a big difference between a Victorian mansion and a mid-century modern home).

CSS PROVIDES EFFICIENCY IN DESIGN AND UPDATES

With CSS, we are able to create rules, and apply those rules to many elements within the website. This approach offers many advantages when site-wide changes are required by a client. Since the content is completely separated from the design, we can make those changes in our Style Sheet and have it effect every applicable instance.

CSS USE CAN LEAD TO FASTER PAGE DOWNLOADS

Since rules are only downloaded once by the browser, then are cached and used for each page load, the use of CSS can lead to lighter page loads, and improved performance. This contributes to lighter server load and lower requirements, which overall saves money for our clients.

CSS IS EASY TO WORK WITH

Because we are able to keep every visual aspect of the website completely separated from the content, using CSS when designing our websites allows us to quickly create layouts, and troubleshoot any problems. We know that regardless of the page, we control all elements from one (or several) stylesheet.

These are just a few reasons why we choose to use CSS. Although we have not gone into the details of using CSS (that is well beyond the scope of this article), we hope that you have a better understanding of why it's important. One last reason is, "all the cool kids are doing it". We're kidding of course, but it is the way of the future…so why not get there now.

1. Separation of Content and Presentation

Cascading Style Sheets are generally located in files separate from the main code (html, for example), permitting a team's different members, such as programmer and designer, to focus on their specialties while working alongside each other, thereby avoiding the risk of interfering with each other's work and affecting the final product.

- Avoid duplication
- Make maintenance easier
- Use the same content with different styles for different purposes

Functional VS non Functional Requirnments

3. Functional requirements is what a system is **supposed to accomplish**. It may be
    - Calculations
    - Technical details
    - Data manipulation
    - Data processing
    - Other specific functionality

While *functional* requirements define what the system does or must not do, *non-functional* requirements specify **how** the system should do it. Non-functional requirements do not affect the basic functionality of the system (hence the name, *non*-functional requirements). Even if the non-functional requirements are not met, the system will still perform its basic purpose.

- Software system requirements are classified as:
- Functional requirements
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations(and sometimes what it should NOT do).
- Non-functional requirements
  - constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc. Apply to the system as whole.

Rest

Rest is way of communication within components on web

Restful Web Service is a lightweight, maintainable, and scalable service that is built on the REST architecture. Restful Web Service, expose API from your application in a secure, uniform, stateless manner to the calling client. The calling client can perform predefined operations using the Restful service. The underlying protocol for REST is HTTP. REST stands for REpresentational State Transfer.

1. **POST** – This would be used to create a new employee using the RESTful web service
2. **GET** - This would be used to get a list of all employee using the RESTful web service
3. **PUT** - This would be used to update all employee using the RESTful web service
4. **DELETE** - This would be used to delete all employee using the RESTful web service

An API for a website is code that allows two software programs to communicate with each other. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application.

SOAP is an XML-based protocol for accessing web services over HTTP. It has some specification which could be used across all applications.

SOAP is known as the Simple Object Access Protocol, but in later times was just shortened to SOAP v1.2. SOAP is a protocol or in other words is a definition of how web services talk to each other or talk to client applications that invoke them.

SOAP was developed as an intermediate language so that applications built on various programming languages could talk easily to each other

• SOAP stands for Simple Object Access Protocol	• REST stands for Representational State Transfer
• SOAP is a protocol. SOAP was designed with a specification. It includes a WSDL file which has the required information on what the web service does in addition to the location of the web service.	• REST is an Architectural style in which a web service can only be treated as a RESTful service if it follows the constraints of being 1. Client Server 2. Stateless 3. Cacheable 4. Layered System 5. Uniform Interface
• SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern.	• REST can make use of SOAP as the underlying protocol for web services, because in the end it is just an architectural pattern.
• SOAP uses service interfaces to expose its functionality to client applications. In SOAP, the WSDL file provides the client with the necessary information which can be used to understand what services the web service can offer.	• REST use Uniform Service locators to access to the components on the hardware device. For example, if there is an object which represents the data of an employee hosted on a URL as http://demo.guru99 , the below are some of URI that can exist to access them http://demo.guru99.com/Employee http://demo.guru99.com/Employee/1

• SOAP requires more bandwidth for its usage. Since SOAP Messages contain a lot of information inside of it, the amount of data transfer using SOAP is generally a lot.
 	
`<?xml version="1.0"?>`
`<SOAP-ENV:Envelope`
	• REST does not need much bandwidth when requests are sent to the server. REST messages mostly just consist of JSON messages. Below is an example of a JSON message passed to a web server. You can see that the size of the message is comparatively smaller to SOAP.
	
	{"city":"Mumbai","state":"Maharastra"}

ARRAY	LINKED LIST
Array is a collection of elements of similar data type.	Linked List is an ordered collection of elements of same type, which are connected to each other using pointers.
Array supports **Random Access**, which means elements can be accessed directly using their index, like `arr[0]` for 1st element, `arr[6]` for 7th element etc. Hence, accessing elements in an array is **fast** with a constant time complexity of `O(1)`.	Linked List supports **Sequential Access**, which means to access any element/node in a linked list, we have to sequentially traverse the complete linked list, upto that element. To access **nth** element of a linked list, time complexity is `O(n)`.
In an array, elements are stored in **contiguous memory location** or consecutive manner in the memory.	In a linked list, new elements can be stored anywhere in the memory. Address of the memory location allocated to the new element is stored in the previous node of linked list, hence formaing a link between the two nodes/elements.
In array, **Insertion and Deletion** operation takes more time, as the memory locations are consecutive and fixed.	In case of linked list, a new element is stored at the first free and available memory location, with only a single overhead step of storing the address of memory location in the previous node of linked list. Insertion and Deletion operations are **fast** in linked list.
Memory is allocated as soon as the array is declared, at **compile time**. It's also known as **Static Memory Allocation**.	Memory is allocated at **runtime**, as and when a new node is added. It's also known as **Dynamic Memory Allocation**.
In array, each element is independent and can be accessed using it's index value.	In case of a linked list, each node/element points to the next, previous, or maybe both nodes.
Array can **single dimensional**, **two dimensional** or **multidimensional**	Linked list can be **Linear(Singly)**, **Doubly** or **Circular** linked list.
Size of the array must be specified at time of array decalaration.	Size of a Linked list is variable. It grows at runtime, as more nodes are added to it.
Array gets memory allocated in the **Stack** section.	Whereas, linked list gets memory allocated in **Heap** section.

AJAX

AJAX stands for **A**synchronous **Ja**vaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

AJAX is a developer's dream, because you can:

- Update a web page without reloading the page
- Request data from a server - after the page has loaded
- Receive data from a server - after the page has loaded
- Send data to a server - in the background

Asynchronous and Synchronous

Synchronous or *Synchronized* means "connected", or "dependent" in some way. In other words, two synchronous tasks must be aware of one another, and one task must execute in some way that is dependent on the other, such as wait to start until the other task has completed. Asynchronous means they are totally independent and neither one must consider the other in any way, either in the initiation or in execution.

STACKS	QUEUES
Stacks are based on the LIFO principle, i.e., the element inserted at the last, is the first element to come out of the list.	Queues are based on the FIFO principle, i.e., the element inserted at the first, is the first element to come out of the list.
Insertion and deletion in stacks takes place only from one end of the list called the top.	Insertion and deletion in queues takes place from the opposite ends of the list. The insertion takes place at the rear of the list and the deletion takes place from the front of the list.
Insert operation is called push operation.	Insert operation is called enqueue operation.
Delete operation is called pop operation.	Delete operation is called dequeue operation.
In stacks we maintain only one pointer to access the list, called the top, which always points to the last element present in the list.	In queues we maintain two pointers to access the list. The front pointer always points to the first element inserted in the list and is still present, and the rear pointer always points to the last inserted element.

Java platform independent

We can say that Java is platform independent because the java interpretor also known as java virtual machine ,converts the java class file into java byte code . This java byte code is machine independent,i.e,it does not depends on which machine it is run. The java compiler converts this java byte code into respective machine code according to the machine it is run on. Since program in java neither depends on software or hardware of the machine it is running, hence it is called platform independent.

```
static int findSum(String str)
{
    // A temporary string
    String temp = "";

    // holds sum of all numbers present in the string
    int sum = 0;

    // read each character in input string
    for(int i = 0; i < str.length(); i++)
    {
        char ch = str.charAt(i);

        // if current character is a digit
        if (Character.isDigit(ch))
            temp += ch;

        // if current character is an alphabet
        else
        {
            // increment sum by number found earlier
            // (if any)
            sum += Integer.parseInt(temp);

            // reset temporary string to empty
            temp = "0";
        }
    }

    // atoi(temp.c_str()) takes care of trailing
    // numbers
    return sum + Integer.parseInt(temp);
}
```

Check power of two

```
static boolean isPowerOfTwo(int n)
{
    if (n == 0)
        return false;

    while (n != 1)
    {
        if (n % 2 != 0)
            return false;
        n = n / 2;
    }
    return true;
}
```

Another

```
if((num != 0) && ((num &(num - 1)) == 0))
   printf("\n%d is a power of 2", num);
```

Binary Search: Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty.

Heap

A heap is a tree-based data structure in which all the nodes of the tree are in a specific order.

For example, if X is the parent node of Y, then the value of X follows a specific order with respect to the value of Y and the same order will be followed across the tree.

Heap data structure is a complete binary tree that satisfies **the heap property**. It is also called as **a binary heap**.

A complete binary tree is a special binary tree in which

- every level, except possibly the last, is filled
- all the nodes are as far left as possible

Heap Property is the property of a node in which

- (for max heap) key of each node is always greater than its child node/s and the key of the root node is the largest among all other nodes;
- (for min heap) key of each node is always smaller than the child node/s and the key of the root node is the smallest among all other nodes.

A linked list is a sequence of data structures, which are connected together via links.

Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.

- **Link** − Each link of a linked list can store a data called an element.
- **Next** − Each link of a linked list contains a link to the next link called Next.
- **LinkedList** − A Linked List contains the connection link to the first link called First.

Rotate array by number k

```
Input arr[] = [1, 2, 3, 4, 5, 6, 7], d = 2, n =7
1) Store the first d elements in a temp array
   temp[] = [1, 2]
2) Shift rest of the arr[]
   arr[] = [3, 4, 5, 6, 7, 6, 7]
3) Store back the d elements
   arr[] = [3, 4, 5, 6, 7, 1, 2]
```

```
LinkedListNode nthToLast(LinkedListNode head, int n) {
  if (head == null || n < 1) {
    return null;
  }

  LinkedListNode p1 = head;
  LinkedListNode p2 = head;

  for (int j = 0; j < n - 1; ++j) { // skip n-1 steps ahead
    if (p2 == null) {
      return null; // not found since list size < n
    }
    p2 = p2.next;
  }

  while (p2.next != null) {
    p1 = p1.next;
    p2 = p2.next;
  }

  return p1;
}
```

Annotations
for form validations in mvc

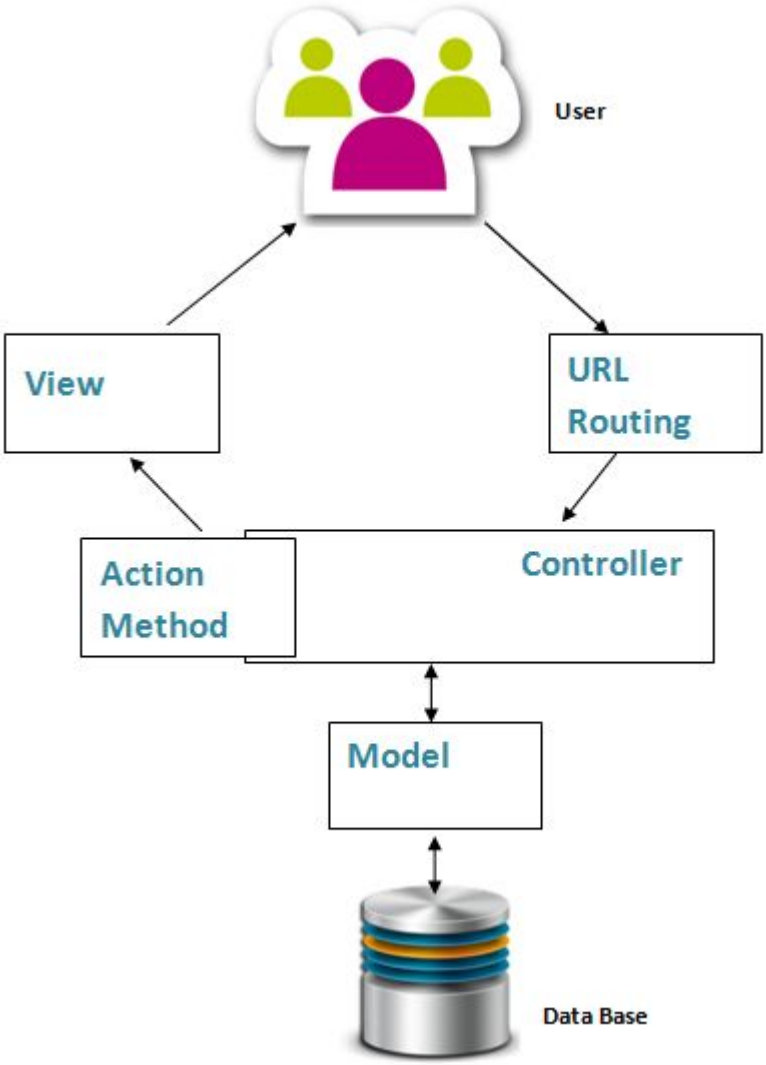
Advantages

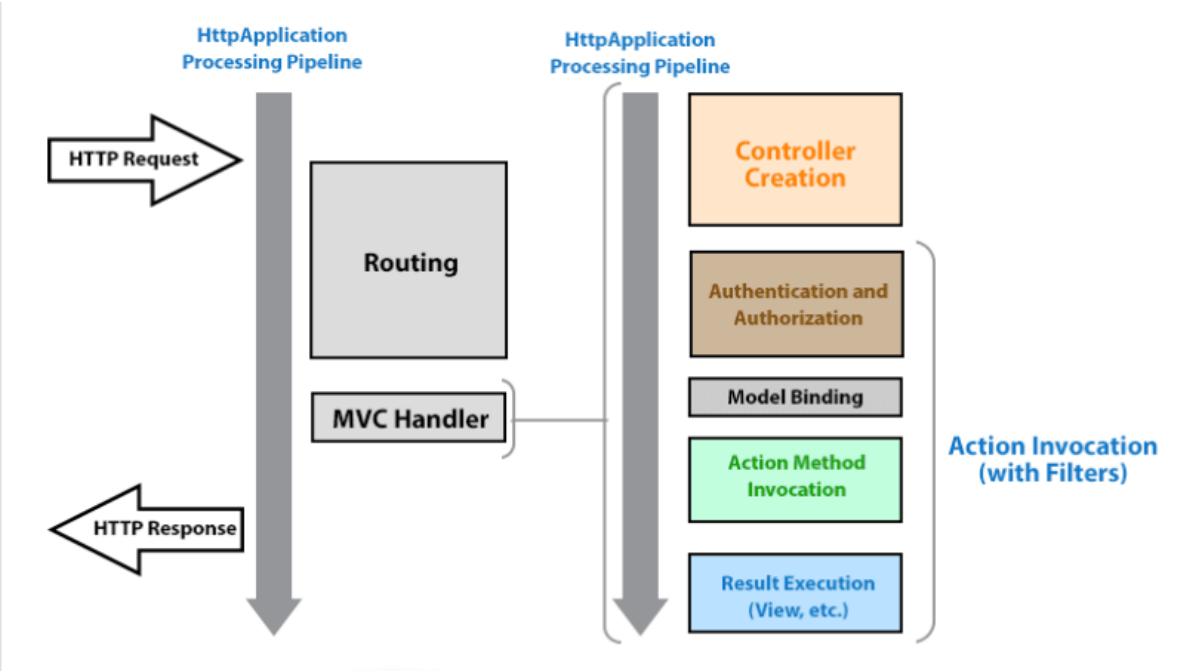
There are many advantages available in MVC.

- A separation of concern is a main advantage of MVC. It means we can divide into three part of the application like Model, View and Controller.
- Handles the code easily because of separation of concern.
- In the same time we can split many developers' work at one time. It will not affect one developer's work to another developer's work.
- It supports TTD (test-driven development). We can create an application with unit test. We can write one test case.
- Latest version of MVC supports default responsive web site and mobile templates.
- We can create our own view engine. Its syntax is very easy compared to traditional view engines.

Disadvantages

- Cannot see design page preview like .aspx page. Every time we want to run it, then we see the design.
- Understanding flow of application is very hard. It is a little bit complex to implement and not suitable for small level applications.
- Its deployment is a little bit hard.





The Bean Validation API is a Java specification which is used to apply constraints on object model via annotations. Here, we can validate a length, number, regular expression, etc. Apart from that, we can also provide custom validations.

@NotNull	It determines that the value can't be null.
@Min	It determines that the number must be equal or greater than the specified value.
@Max	It determines that the number must be equal or less than the specified value.
@Size	It determines that the size must be equal to the specified value.
@Pattern	It determines that the sequence follows the specified regular expression.

```
package com.javatpoint;
import javax.validation.constraints.Size;

public class Employee {

   private String name;
   @Size(min=1,message="required")
   private String pass;
```

```
@NotNull
@Size(min=2, max=30)
private String name;

@NotNull
@Min(18)
private Integer age;
```

Difference between User-Level & Kernel-Level Thread

S.N.	User-Level Threads	Kernel-Level Thread
1	User-level threads are faster to create and manage.	Kernel-level threads are slower to create and manage.
2	Implementation is by a thread library at the user level.	Operating system supports creation of Kernel threads.
3	User-level thread is generic and can run on any operating system.	Kernel-level thread is specific to the operating system.
4	Multi-threaded applications cannot take advantage of multiprocessing.	Kernel routines themselves can be multithreaded.