1) what is java?
==> Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. ... Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.
The types of the Java programming language are divided into two categories: primitive types and reference types. The primitive types (§4.2) are the boolean type and the numeric types. The numeric types are the integral types byte, short, int, long, and char, and the floating-point types float and double.
The reference types (§4.3) are class types, interface types, and array types.

2) Why java is platform independent?
==>Java compiler converts Java code in to byte code that can be interpreted by JVM.
The most unique feature of java is platform independent. In any programming language soruce code is compiled in to executable code . This cannot be run across all platforms. When javac compiles a java program it generates an executable file called .class file.
class file contains byte codes. Byte codes are interpreted only by JVM's . Since these JVM's are made available across all platforms by Sun Microsystems, we can execute this byte code in any platform. Byte code generated in windows environment can also be executed in linux environment. This makes java platform independent.

3) What is Java Virtual Machine(JVM)?
==>Java Virtual Machine (JVM) is an abstract machine that executes Java Bytecode. There are different JVM for different hardware and software platforms. So JVM is platform dependent. JVM is responsible for loading, verifying and executing the Bytecode on a platform.

4) What is the difference between JDK and JRE?
==> JDK stands for Java Development Kit. It contains the tools and libraries for development of Java programs. It also contains compilers and debuggers needed to compile Java program,

==>JRE stands for Java Runtime Environment. This is included in JDK.
JRE provides libraries and JVM that is required to run a Java program

========Oops==========

1)1)Constrctor:
==>In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.

2)Types of Constrctor Chaning:
==>1)Default constructor (no-arg constructor)
2)Parameterized constructor

1)Default constructor (no-arg constructor):
==>The default constructor is used to provide the default values to the object like 0, null, etc., depending on the type.

2)Parameterized constructor :
==>A constructor which has a specific number of parameters is called a parameterized constructor.

3)Why use the parameterized constructor?
The parameterized constructor is used to provide different values to distinct objects. However, you can provide the s

ame values also.

4)Constructor chaining :
==>is the process of calling one constructor from another constructor with respect to current object.

5)==>object : Object means a real-world entity such as a pen, chair, table, computer, watch, etc.
==>oops: Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts:
Object
Class
Inheritance
Polymorphism
Abstraction
Encapsulation

==>object : Any entity that has state and behavior is known as an object.
For example, a chair, pen, table, keyboard, bike, etc. It can be physical or logical.
eg: A dog is an object because it has states like color, name, breed, etc.
as well as behaviors like wagging the tail, barking, eating, etc.

==>class : Collection of objects is called class. It is a logical entity.
A class can also be defined as a blueprint from which you can create an individual object. Class doesn't consume any space.

==>Inheritance : When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.
example, the relationship between father and son.

==>Why use inheritance in java
1)For Method Overriding (so runtime polymorphism can be achieved).
2)For Code Reusability.

==>Terms used in Inheritance
1)Class: A class is a group of objects which have common properties.
It is a template or blueprint from which objects are created.

2)Sub Class/Child Class: Subclass is a class which inherits the other class.
It is also called a derived class, extended class, or child class.

3)Super Class/Parent Class: Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.

4)Reusability: As the name specifies, reusability is a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class.
You can use the same fields and methods already defined in the previous class.

==>Polymophism : If one task is performed in different ways, it is known as polymorphism.
For example: to convince the customer differently, to draw something, for example, shape, triangle, rectangle, etc

==>Abstarction : Hiding internal details and showing functionality is known as abstraction.
For example phone call, we don't know the internal processing.
e.g:Consider a real-life example of a man driving a car. The man onlyknows that pressing the accelerators will increase the speed of car or applying brakes will stop the car, but he does not know about how on pressing the accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the

implementation of the accelerator, brakes, etc in the car

==>Encapsulation : Binding (or wrapping) code and data together into a single unit are known as encapsulation.
For example, a capsule, it is wrapped with different medicines

==>Method overloading : If a class has multiple methods having same name but different in parameters,
it is known as Method Overloading.

8)What is method overloading in java ?
==>A class having two or more methods with same name but with different arguments then we say that those methods are overloaded.

```
e.g: class Adder{
static int add(int a,int b){return a+b;}
static int add(int a,int b,int c){return a+b+c;}
}
class TestOverloading1{
public static void main(String[] args){
System.out.println(Adder.add(11,11));
System.out.println(Adder.add(11,11,11));
}}
```

==>Method overriding : If subclass (child class) has the same method as declared in the parent class,
it is known as method overriding in Java.

What is method overriding in java ?
==>If we have methods with same signature (same name, same signature, same return type) in super class and subclass then we say subclass method is overridden by superclass.
When to use overriding in java If we want same method with different behaviour in superclass and subclass then we go for overriding.
When we call overridden method with subclass reference subclass method is called hiding the superclass method.

```
e.g:class Vehicle{
  //defining a method
  void run(){System.out.println("Vehicle is running");}
}
//Creating a child class
class Bike2 extends Vehicle{
  //defining the same method as in the parent class
  void run(){System.out.println("Bike is running safely");}

  public static void main(String args[]){
  Bike2 obj = new Bike2();//creating object
  obj.run();//calling method
  }
}
```

==>Wrapper class : The wrapper class in Java provides the mechanism to convert primitive into object and object into primitive.
Since J2SE 5.0, autoboxing and unboxing feature convert primitives into objects and objects into primitives automatically. The automatic conversion of primitive into an object is known as autoboxing and vice-versa unboxing
.

==>recursion : Recursion in java is a process in which a method calls itself continuously.
A method in java that calls itself is called recursive method

==>diff obj and class :

object	class
1)Object is an instance of a class.	1)Class is a blueprint or template from which objects are created.
2)Object is a real world entity such as pen, laptop, mobile, bed, keyboard, mouse, chair etc.	2)Class is a group of similar objects.
3)Object is a physical entity.	3)Class is a logical entity.
4)Object is created many times as per requirement.	4)Class is declared once.
5)Object allocates memory when it is created.	5)Class doesn't allocated memory when it is created.

==> diff overriding and overloading

overloading	overriding
1)Method overloading is performed within class.	1)Method overriding occurs in two classes that have IS-A (inheritance) relationship.
2)In case of method overloading, parameter must be different.	2)In case of method overriding, parameter must be same.
3)Method overloading is the example of compile time polymorphism.	3)Method overriding is the example of run time polymorphism.

==>Didderence abstract class and interface

Abstract class	Interface
1) Abstract class can have abstract and non-abstract methods.	1)Interface can have only abstract methods. Since Java 8, it can have default and static methods also.
2) Abstract class doesn't support multiple inheritance.	2)Interface supports multiple inheritance.
3) Abstract class can have final, non-final, static and non-static variables.	3)Interface has only static and final variables.
4) Abstract class can provide the implementation of interface.	4)Interface can't provide the implementation of abstract class.
5) The abstract keyword is used to declare abstract class.	5)The interface keyword is used to declare interface.
6) An abstract class can be extended using keyword "extends".	6)An interface can be implemented using keyword "implements".
7) A Java abstract class can have class members like private, protected, etc.	7)Members of a Java interface are public by default.

==>Difference c++(cpp) and java

cpp	java
1)C++ is platform-dependent.	1)Java is platform-independent.
2)C++ is mainly used for system programming.	2)Java is mainly used for application programming. It is widely used in Windows-based, web-based, enterprise, and mobile applications.
3)C++ supports the goto statement.	3)Java doesn't support the goto statement.
4)C++ supports multiple inheritance.	4)Java doesn't support multiple inheritance
5)C++ supports both call by value and call by reference.	5)Java supports call by value only. There is no call by reference in java.

==>Difference bet multiple and multilevel inheritance

Multiple inheritance	Multilevel inheritance
1)we can inherits more than one class in the same class.	1)where one class can inherits only one base class & the derived class can become base class of

some other class

2)e.g:a class defining child inherits from two base class mother & father

2)e.g:a class describibg a sports car will inherits from a base class car which inherits another class vehical

==>difference bet Error and Exception

Error	Exception
1)Classified as an unchecked type	1)Classified as checked and unchecked
2)It belongs to java.lang.error	2)It belongs to java.lang.Exception
3)It is irrecoverable	3)It is recoverable
4)It can't be occur at compile time	4)It can occur at run time compile time both
5)e.g:OutOfMemoryError ,IOError	5)e.g:NullPointerException , SqlException

==>Difference bet Comparable and Comparator

Comparable	Comparator
1)Comparable provides a single sorting sequence.In other words, we can sort the collection on the basis of a single element such as id, name, and price.	1)The Comparator provides multiple sorting sequences. In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc.
2)omparable affects the original class, i.e., the actual class is modified.	2)Comparator doesn't affect the original class, i.e., the actual class is not modified.
3)Comparable provides compareTo() method to sort elements.	3)Comparator provides compare() method to sort elements.
4)Comparable is present in java.lang package.	4)A Comparator is present in the java.util package.

===>Difference bet Array & Arraylist

Array	ArrayList
1)An array can store both objects and primitives type.	1)We cannot store primitive type in ArrayList. It automatically converts primitive type to object.
2)Array is static in size.	2)ArrayList is dynamic in size.
3)An array is a fixed-length data structure.	3)ArrayList is a variable-length data structure.
4)ray can be multi-dimensional.	4)ArrayList is always single-dimensional.
5)Array provides a length variable which denotes the length of an array.	5)ArrayList provides the size() method to determine the size of ArrayList.

==>Difference bet Process and Thread

Process	Thread
1)A process is a program under execution i.e an active program.	1)A thread is a lightweight process that can be managed independently
2)Processes require more time for context switching	2)Threads require less time for context switching
3)Processes require more time for creation.	3)Threads require less time for creation.
4)Processes require more time for termination.	4)Threads require less time for termination.
5)Communication between processes requires more time than between threads.	5)Communication between threads requires less time than between processes .
6)process is heavy weight	6)thread is light weight

==>Difference bet Hashmap & Hashtable

Hashmap	Hashtable
1)HashMap is not synchronized.	1)Hashtable is synchronized.
2)HashMap allows atmost one null key and any number of null values.	2)Hashtable does not allow null values.
3)Since HashMap is not synchronized its performance is faster than Hashtable	3)Performance is slower when compared to HashMap.

4)HashMap introduced starting from collection framework	4)Hashtable is even before collection framework

==>Difference Arraylist & Linkedlist

Arraylist	Linkedlist
1)Implements RandomAccess interface we can search randomly all the elements in the list.	1)It extends Abstract sequential List interface which provides sequential access to elements
2)Adding and removal of elements in random positions is slow.	2)Adding and removal of elements in random positions is fast.

==>Access modifiers :The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

There are four types of Java access modifiers:

1)Private: The access level of a private modifier is only within the class.
It cannot be accessed from outside the class.

2)Default: The access level of a default modifier is only within the package.
It cannot be accessed from outside the package. If you do not specify any access level, it will be the default.

3)Protected: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.

4)Public: The access level of a public modifier is everywhere.
It can be accessed from within the class, outside the class, within the package and outside the package.

==>Instance Variable :
They are defined in class but outside the body of methods.
==>Local Variable :
They are defined as a type of variable declared within programming blocks or subroutines.

==>What is the purpose of public static void main String args?
1)Public- it is access specifier from anywhere we can access it
2)Static- it is access modifier we can call the methods directly by class name without creating its objects
3)Void- it is the return type
4)Main- it is a method name
5)String[]args- in java we accept only the string type of argument and store it

5)What is a finally block in Java?
==>Java provides a finally block with a try block. This is an optional block. But finally block is always executed after the execution of try block.

======Excepetion handling==========

6)What is an exception in java?
In java exception is an object. Exceptions are created when an abnormal situations are arised in our program. Exceptions can be created by JVM or by our application code. All Exception classes are defined

in java.lang. In otherwords we can say Exception as run time error.

7).What is Exception Handling in Java?
==> Java provides Exception Handling mechanism to handle Runtime
errors that occur in JVM. There are checked exceptions in a
program that we expect to occur in certain situations.
Exception handling mechanism catches these checked exceptions and takes relevant actions.

8)What is an eror in Java?
Error is the subclass of Throwable class in java. When errors are caused by our program we call that as Exception, but some times exceptions are caused due to some environment issues such as running out of memory. In such cases we can't handle the exceptions. Exceptions which cannot be recovered are called as errors in java.
Ex : Out of memory issues.

9)What are advantages of Exception handling in java?
==>1) Separating normal code from exception handling code to avoid abnormal termination of program.
2) Categorizing in to different types of Exceptions so that rather than handling all exceptions with Exception root class we can handle with specific exceptions. It is recommended to handle exceptions with specific Exception instead of handling with Exception root class.
3) Call stack mechanism : If a method throws an exception and it is not handled immediately, then that exception is propagated or thrown to the caller of that method. This propogation continues till it finds an appropriate exception handler ,if it finds handler it would be handled otherwise program terminates abruptly.

10) In how many ways we can do exception handling in java?
==>We can handle exceptions in either of the two ways :
1) By specifying try catch block where we can catch the exception.
2) Declaring a method with throws clause .

11) List out five keywords related to Exception handling ?
==>1) Try
2) Catch
3) throw
4) throws
5) finally

12)59) Explain try and catch keywords in java?
==>In try block we define all exception causing code. In java try and catch forms a unit. A catch block catches the exception thrown by preceding try block. Catch block cannot catch an exception thrown by another try block. If there is no exception causing code in our program or exception is not raised in our code jvm ignores the try catch block.
Syntax :
try
{}
Catch(Exception e)
{}

13) Can we have try block without catch block?
==>Each try block requires atleast one catch block or finally block. A try block without catch or finally will result in compiler error. We can skip either of catch or finally block but not both.

14)Explain importance of finally block in java?
==>Finally block is used for cleaning up of resources such as closing connections, sockets etc. if try block executes with no exceptions then finally is called after try block without executing catch block. If there is

exception thrown in try block finally block executes immediately after catch block.
If an exception is thrown,finally block will be executed even if the no catch block handles the exception.

15)What are checked Exceptions?
==>1) All the subclasses of Throwable class except error,Runtime Exception and its subclasses are checked exceptions.
2) Checked exception should be thrown with keyword throws or should be provided try catch block, else the program would not compile. We do get compilation error.
==>Examples :
1) IOException,
2) SQlException,
3) FileNotFoundException,
4) InvocationTargetException,
5) CloneNotSupportedException
6) ClassNotFoundException
7) InstantiationException

16) What are unchecked exceptions in java?
==>All subclasses of RuntimeException are called unchecked exceptions. These are unchecked exceptions because compiler does not checks if a method handles or throws exceptions.
Program compiles even if we do not catch the exception or throws the exception.
If an exception occurs in the program,program terminates . It is difficult to handle these exceptions because there may be many places causing exceptions.
==>Example : 1) Arithmetic Exception
3) ArrayIndexOutOfBoundsException
4) ClassCastException
5) IndexOutOfBoundException
6) NullPointerException
7) NumberFormatException
8) StringIndexOutOfBounds
9) UnsupportedOperationException

17)Explain differences between checked and Unchecked exceptions in java?

Unchecked Exception	Checked Exception
1)All the subclasses of RuntimeException are called unchecked exception.	1)All subclasses of Throwable class except RuntimeException are called as checked exceptions
2)Unchecked exceptions need not be handled at compile time	2)Checked Exceptions need to be handled at compile time.
3)ArrayIndexOutOfBoundsException, ClassCastException, IndexOutOfBoundException	3)SqlException, FileNotFoundException,ClassNotFoundException

18)Explain throw keyword in java?
==>Generally JVM throws the exception and we handle the exceptions by using try catch block. But there are situations where we have to throw userdefined exceptions or runtime exceptions. In such case we use throw keyword to throw exception explicitly.
Syntax : throw throwableInstance;

19)Explain importance of throws keyword in java?
==>Throws statement is used at the end of method signature to indicate that an exception of a given type may be thrown from the method.
The main purpose of throws keyword is to delegate responsibility of exception handling to the caller methods, in the case of checked exception.

10)What is super keyword in java ?
==>Variables and methods of super class can be overridden in subclass . In case of overriding , a subclass

object call its own variables and methods. Subclass cannot access the variables and methods of superclass because the overridden variables or methods hides the methods and variables of super class. But still java provides a way to access super class members even if its members are overridden. Super is used to access superclass variables, methods, constructors.
Super can be used in two forms :
1) First form is for calling super class constructor.
2) Second one is to call super class variables,methods.
Super if present must be the first statement.

=========Thread=======

1)What is process ?
==>A process is a program in execution.
Every process have their own memory space.Process are heavy weight and requires their own address space. One or more threads make a process.

2)What is the thread?
==>A thread is a lightweight subprocess. It is a separate path of execution because each thread runs in a different stack frame.

3)What is multithreading?
==>Multithreading is a process of executing multiple threads simultaneously. Multithreading is used to obtain the multitasking. It consumes less memory and gives the fast and efficient performance. Its main advantages are:

Threads share the same address space.
The thread is lightweight.
The cost of communication between the processes is low.

4) What is multitasking ?
==>Multitasking means performing more than one activity at a time on the computer. Example Using spreadsheet and using calculator at same time.

5)What are different types of multitasking?
==>There are two different types of multitasking :
1) Process based multitasking
2) Thread based multitasking
Process based multitasking : It allows to run two or more programs concurrently. In process based multitasking a process is the smallest part of code .
Example : Running Ms word and Ms powerpoint at a time.
Thread based multitasking : It allows to run parts of a program to run concurrently.
Example : Formatting the text and printing word document at same time .
Java supports thread based multitasking and provides built in support for multithreading.

6)List Java API that supports threads?
==>java.lang.Thread : This is one of the way to create a thread. By extending Thread class and overriding run() we can create thread in java.
java.lang.Runnable : Runnable is an interface in java. By implementing runnable interface and overriding run() we can create thread in java.
java.lang.Object : Object class is the super class for all the classes in java. In object class we have three methods wait(), notify(), notifyAll() that supports threads.
java.util.concurrent : This package has classes and interfaces that supports concurrent programming.
Ex : Executor interface, Future task class etc.

7)Explain the life cycle of thread?
==>A thread can be in any of the five states :

1) New : When the instance of thread is created it will be in New state.
Ex : Thread t= new Thread();
In the above example t is in new state. The thread is created but not in active state to make it active we need to call start() method on it.
2) Runnable state : A thread can be in the runnable state in either of the following two ways :
a) When the start method is invoked or
b) A thread can also be in runnable state after coming back from blocked or sleeping or waiting state.
3) Running state : If thread scheduler allocates cpu time, then the thread will be in running state.
4) Waited /Blocking/Sleeping state:
In this state the thread can be made temporarily inactive for a short period of time. A thread can be in the above state in any of the following ways:
1) The thread waits to acquire lock of an object.
2) The thread waits for another thread to complete.
3) The thread waits for notification of other thread.
5) Dead State : A thread is in dead state when thread's run method execution is complete. It dies automatically when thread's run method execution is completed and the thread object will be garbage collected.

8)In how many ways we can do synchronization in java?
==>There are two ways to do synchronization in java:
1) Synchronized methods
2) Synchronized blocks
To do synchronization we use synchronize keyword.

9) What are synchronized methods ?
==>If we want a method of object to be accessed by single thread at a time we declare that method with synchronized keyword.
Signature :
public synchronized void methodName(){}

10) What are synchronized blocks in java?
==>Synchronizing few lines of code rather than complete method with the help of synchronized keyword are called synchronized blocks.
Signature :
Synchronized (object reference){// code}

11) What is class level lock ?
==>Acquiring lock on the class instance rather than object of the class is called class level lock. The difference between class level lock and object level lock is in class level lock lock is acquired on class .class instance and in object level lock ,lock is acquired on object of class.

12)What all methods are used to prevent thread execution ?
==>There are three methods in Thread class which prevents execution of thread.
1) yield()
2) join()
3) sleep()

12) Explain yield() method in thread class ?
==>Yield() method makes the current running thread to move in to runnable state from running state giving chance to remaining threads of equal priority which are in waiting state.

13)What is the purpose of wait() method in Java?
==>The wait() method is provided by the Object class in Java. This method is used for inter-thread communication in Java. The java.lang.Object.wait() is used to pause the

current thread, and wait until another thread does not call the notify() or notifyAll() method.

14) What does join() method?
==>The join() method waits for a thread to die. In other words, it causes the currently running threads to stop executing until the thread it joins with completes its task.

15) What does sleep() method?
==>The sleep() method in java is used to block a thread for a particular time, which means it pause the execution of a thread for a specific time.

16)What is the difference between notify() and notifyAll()?
==>The notify() is used to unblock one waiting thread whereas notifyAll() method is used to unblock all the threads in waiting state.

17)What is the deadlock?
==>Deadlock is a situation in which every thread is waiting for a resource which is held by some other waiting thread. In this situation, Neither of the thread executes nor it gets the chance to be executed. Instead, there exists a universal waiting state among all the threads. Deadlock is a very complicated situation which can break our code at runtime.

====Collection=======

1)What is collections framework ?
==>A framework is set of classes and interfaces to build a functionality.
Java collections framework provides
set of interfaces and classes for storing and manipulating collections. Collection framework contains classes and interfaces in java.util package and java.util.concurrent packages.
Advantages or benefits of Collections framework :
1) High performance
2) Using this framework we can create different types of collections
3) We can create our own collection and we can extend a collection.
4) Reduces programming effort.
5) Increases speed and quality : Collections framework provides high performance, implementations of
useful data structures and algorithms.

2)What is collection ?
==>A collection is a container which holds group of objects. Collection provides a way to manage objects easily. Collections manages group of objects as single unit. Examples include list of strings, integers etc.

3) Difference between collection, Collection and Collections in java?
==>collection : represent group of objects where objects are stored.
Collection : This is one of the core interface which provides basic functionality for collection.
Collections : Collections contains some utility static methods that operate on collections.

4) List the interfaces which extends collection interface ?
1) List
2) Set
3) Queue
4) Deque

5) Explain List interface ?
==>List interface extends collection interface used to store sequence of elements in collection.
We can even store duplicate elements in list.
We can insert or access elements in list by using index as we do in arrays.

6) List implementations of List Interface ?
==>1) ArrayList
2) Vector
3) LinkedList

7) Explain about ArrayList ?
==>ArrayList is an ordered collection which extends AbstractList and implements List interface.
We use ArrayList mainly when we need faster access and fast iteration of elements in list.
We can insert nulls in to arraylist.
Arraylist is nothing but a growable array.
List is an ordered collection.

8)What is vector?
==>Vector is similar to arraylist used for random access.
Vector is a dynamic array like arraylist.
vector size increases or decreases when elements are added and removed .
Vector is synchronized .

====Serialization==========
1)What is serialization in java?
==>Serialization is the process of converting an object in to bytes, so that it can be transmitted over the network,or stored in a flat file and can be recreated later. Serialized object is an object represented as sequence of bytes that includes objects data, object type, and the types of data stored in the object.

2) What is the main purpose of serialization in java?
==>The main uses of serialization are :
1) Persistence:
We can write data to a file or database and can be used later by deserializing it.
2) Communication :
To pass an object over network by making remote procedure call.
3) Copying :
We can create duplicates of original object by using byte array.
4) To distribute objects across different JVMs.

===>Data structure ==>

==>data structure : The data structure is a way that specifies how to organize and manipulate the data. It also defines the relationship between them.
Some examples of Data Structures are arrays, Linked List, Stack, Queue, etc.
Data Structures are the central part of many computer science algorithms as they enable the programmers to handle the data in an efficient way.

==>types of ds :
1)Primitive Data structure : The primitive data structures are primitive data types.
The int, char, float, double, and pointer are the primitive data structures that can hold a single value.

non -primitive:
2)Linear Data Structure: A data structure is called linear if all of its elements

are arranged in the sequential order. In linear data structures, the elements are stored in a non-hierarchical way where each item has the successors and predecessors except the first and last element.
The data structures used for this purpose are Arrays, Linked list, Stacks, and Queues. In these data structures, one element is connected to only one another element in a linear form.

3)non linear :When one element is connected to the 'n' number of elements known as a non-linear data structure. The best example is trees and graphs.
In this case, the elements are arranged in a random manner.

==>Stack :
A Stack is a linear data structure that follows the LIFO (Last-In-First-Out) principle. A stack can be defined as a container in which insertion and deletion can be done from the one end known as the top of the stack.

● push(): When we insert an element in a stack then the operation is known as a push. If the stack is full then the overflow condition occurs.
● pop(): When we delete an element from the stack, the operation is known as a pop. If the stack is empty means that no element exists in the stack, this state is known as an underflow state.
● isEmpty(): It determines whether the stack is empty or not.
● isFull(): It determines whether the stack is full or not.'
● peek(): It returns the element at the given position.
● count(): It returns the total number of elements available in a stack.
● change(): It changes the element at the given position.
● display(): It prints all the elements available in the stack.
eg:● Back/Forward on browsers are perform using stacks.

==>Queue:
A queue in the data structure can be considered similar to the queue in the real-world.
A queue is a data structure in which whatever comes first will go out first.
It follows the FIFO(First-In-First Out) policy.
==In Queue, the insertion is done from one end known as the rear end or the tail of the queue, whereas the deletion is done fromanother end known asthe front end or the headof the queue.
In other words, it can be defined as a list or a collection with a constraint that the insertion can be performed at one end called as the rear end or tail of the queueand deletion is performed on another end called as the front end or the head of the queueTypes of Queue
eg: Vehicles on toll-tax bridge: The vehicle that comes first to the toll tax booth leaves the booth first. The vehicle that comes last leaves last. Therefore, it follows first-in-first-out (FIFO)

1)Liner Queue: In Linear Queue, an insertion takes place from one end while the deletion occurs from another end. The end at which the insertion takes place is known as the rear end, and the end at which the deletion takes place is known as front end.

2)circule Queue: In Circular Queue, all the nodes are represented as circular. It is similar to the linear Queue except that the last element of the queue is connected to the first element. It is also known as Ring Buffer as all the ends are connected to another end.

3)Priority Queue: A priority queue is another special type of Queue data structure in which each element has some priority associated with it. Based on the priority of the element, the elements are arranged in a priority queue. If the elements occur with the same priority, then they are served according to the FIFO principle.

4)decue :Both the Linear Queue and Deque are different as the linear queue follows the FIFO principle

whereas, deque does not follow the FIFO principle. In Deque, the insertion anddeletion can occur from both ends.

==>Linked list ~: Linked List is a type of linear data structure.in linked list data is not stored in contiguous location. Linked List consists of nodes where each node contains data and link to the nextnode.
·In nodes there are two sections which contain data and address of the next node so withthe help of address we access data in the linked list.
·There are 3 types of linked list – 1. Singly linked list 2. Doubly linked list 3. Circular linkedlist
1) Singly linked list – in singly linked list each node contains data and address of nextnode so we can traverse in only one direction.
eg: ticket counter

2)Doubly linked list – in doubly linked list each node contains data and address of previous and next node. So we can traverse in forward as well as backward direction.
eg: A music player which has next and previous buttons.

3)Circular linked list – in circular linked list tail is connected to head. A circular linked list can be a Singly circular linked list or doubly circular linked list.
eg: our Personal Computers,

==>Hash table : A hash table(hash map) is a type of data structure which is used forstoring and accessing data very quickly.

==>Hash function:- It is used to generate the corresponding value to store the key.It isbasically a mathematical function known as Hash function

==>Binary search tree: Binary Search Tree is also known as an ordered tree , it is a tree which contains the lesser numberon the left hand side of tree and greater number on the right hand side.The left and right subtree each also be a binary search tree.

==>Tree: A tree is a Hierarchical data structure that naturally hierarchically stores the information. The Tree data structure is one of the most efficient and mature.The nodes connected by the edges are represented.

==>Types of tree :
1. General Tree: If no constraint is placed on the tree's hierarchy, a tree is called a general tree. Every node may have infinite numbers of children in General Tree. The tree is the super-set of all other trees.

2. Binary Tree
The binary tree is the kind of tree in which most two children can be found foreach parent. The kids are known as the left kid and right kid. This is more popular than most other trees.

3. Binary Search Tree
Binary Search Tree (BST) is a binary tree extension with several optional restrictions. The left child value of a node should in BST be less than or equal to the parent value, and the right child value should always be greater than orequal to the parent's value.

4.Avl tree : AVL Tree can be defined as height balanced binary search tree in which each node is associated with a balance factor which is calculated by subtracting the height of its rightsub-tree from that of its left sub-tree

==>Sort
==>sorting : Sorting refers to arranging data in a particular format.

Sorting algorithm specifies the way to arrange data in a particular order.
Sorting is the process of arranging the elements of an array so that they can be placed either in ascending or descending order.

e.g : Telephone Directory − The telephone directory stores the telephone numbers of people sorted by their names, so that the names can be searched easily.

Dictionary − The dictionary stores words in an alphabetical order so that searching of any word becomes easy.

1)Bubble sort : Bubble sort is a simple sorting algorithm. This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order. This algorithm is not suitable for large data sets as its average and worst case complexity are of O(n2) where n is the number of items.

2)Bucket sort : Bucket sort is also known as bin sort. In the Bucket Sorting technique, the data items are distributed in a set of buckets. Each bucket can hold a similar type of data. After distributing, each bucket is sorted using another sorting algorithm.

3)Insertion sort : Insertion sort works similar to the sorting of playing cards in hands. It is assumed that the first card is already sorted in the card game, and then we select an unsorted card. If the selected unsorted card is greater than the first card, it will be placed at the right side; otherwise, it will be placed at the left side. Similarly, all unsorted cards are taken and put in their exact place.
The same approach is applied in insertion sort. The idea behind the insertion sort is that first take one element, iterate it through the sorted array.

4)Heap sort : Heap sort processes the elements by creating the min heap or max heap using the elements of the given array. Min heap or max heap represents the ordering of the array in which root element represents the minimum or maximum element of the array.

==MySql ==>

joins :As the name shows, JOIN means to combine something. In case of SQL, JOIN means "to combine two or more tables".
The SQL JOIN clause takes records from two or more tables in a database and combines it together

1)inner join :Inner Join is the simplest and most common type of join. It is also known as simple join. It returns all rows from multiple tables where the join condition is met

2).Left outer join (also known as left join): this join returns all the rows from left table combine with the matching rows of the right table. If you get no matching in the right table it returns NULL values.

3).Right outer join (also known as right join): this join returns all the rows from right table are combined with the matching rows of left table .If you get no column matching in the left table . it returns null value.

4)full join : SQL full outer join is used to combine the result of both left and right outer join and returns all rows (don't care its matched or unmatched) from the both participating tables.

5)self join: Self Join is a specific type of Join. In Self Join, a table is joined with itself (Unary relationship). A self join simply specifies that each rows of a table is combined with itself and every other row of the table.

6)cross join :The CROSS JOIN specifies that all rows from first table join with all of the rows of second table.

==>ACID property :
Atomacitry: A transaction must be an atomic unit of work, which means that all the modified data are performed or none of them will be.

Consistency :This property ensures that the transaction maintains data integrity constraints, leaving the data consistent.

Isolation :This property ensures the isolation of each transaction, ensuring that the transaction will not be changed by any other concurrent transaction.

Durability : Once a transaction is completed and committed, its changes are persisted permanently in the database.

==>Diff rdbms and dbms

No.	DBMS	RDBMS
1)	DBMS applications store data as file.	RDBMS applications store data in a tabular form.
3)	Normalization is not present in DBMS.	Normalization is present in RDBMS.
7)	DBMS does not support distributed database.	RDBMS supports distributed database.
9)	Examples of DBMS are file systems, xml etc.	Example of RDBMS are mysql, postgre, sql server, oracle etc.
	DBMS is meant to be for small organization and deal with small data. it supports single user.	RDBMS is designed to handle large amount of data. it supports multiple users.

==>database: The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

==>rdbms :A Relation Database Management system (RDBMS) is a database management system that is based on the relational model. It has the following major components: Table, Record/Tuple/Row, Field, and Column/Attribute. Examples of the most popular RDBMS are MYSQL, Oracle, IBM DB2, and Microsoft SQL Server database

==>DDL – Data Definition Language (create , drop, alter, trancate, commit , rollback~~)
==>DML Data Manipulation Language (insert, update,delete)
==>DCL – Data Control Language (grant, revok)
==>DQl – Data Query Language(select)

==>What is Normalization :-Normalization is the process of organizing the data in the database. Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships.

1)1NF (First Normal Form) Rules
Each table cell should contain a single value.
Each record needs to be unique.

2)Second Normal Form (2NF)

In the 2NF, relational must be in 1NF.
In the second normal form, all non-key attributes are fully functional dependent on the primary key

==>Key :Keys play an important role in the relational database.
It is used to uniquely identify any record or row of data from the table.
It is also used to establish and identify relationships between tables.

1)Primary key : The PRIMARY KEY constraint uniquely identifies each record in a table.
Primary keys must UNIQUE values, and cannot contain NULL values.
A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

2)Forgin key:The FOREIGN KEY is used to prevent actions that would destroy links between tables.A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

3)uniqe key :The UNIQUE constraint ensures that all values in a column are different.
Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.
A PRIMARY KEY constraint automatically has a UNIQUE constraint.

===> Project info ====>

Rentaru is a self-drive car rental service that allows users to rent cars.
This Car Rental System project is designed to rent the car for the specified period through an online system. It helps the users to search for available cars view profiles and book the cars for the time period. They could also make payments online.
Based on the type of car required by the customer, the user shall be able to make bookings.
This Car Rental System makes bookings easy. It saves time and labor.

===Spring Boot=======

1)customer.java (Entity pacakage)
1) @Entity : Entity is annotation specifies that the class is an entity and is mapped to a database table.

2) @Table :annotation specifies the name of the database table to be used for mapping.

3) @GeneratedValue : Provides for the specification of generation strategies for the values of primary keys.
For example : when using Mysql , you may specify auto_increment in the definition of table to make it self-incremental, and then use @GeneratedValue(strategy = GenerationType.IDENTITY)

4) @Column: The Column annotation is used to specify the mapped column for a persistent property or field. If no Column annotation is specified, the default value will be applied.
The @Column(nullable = false) : annotation only adds a not null constraint to the table definition.
Hibernate or any other framework will not perform any validation on the entity attribute. ...
If the entity attribute is null, the SQL statement will fail

5) @NotNull : annotation is, actually, an explicit contract declaring that:
A method should not return null. Variables (fields, local variables, and parameters) cannot hold a null value.

6) @Size : annotation is used to restrict the filed length to a specified value.

It has attributes such as max and min which are used to set the maximum and minimum values respectively. The message attribute in this annotation is used to display a default message on validation failure.

7) @Temporal : is a JPA annotation which can be used to store in the database table on of the following column items: DATE (java. TIME (java. sql. Time)

2)CustomerRepository(Repository package):
8) JpaRepository : is JPA specific extension of Repository . It contains the full API of CrudRepository and PagingAndSortingRepository . So it contains API for basic CRUD operations and also API for pagination and sorting.

9) @Modifying annotation is used to enhance the @Query annotation to execute not only SELECT queries but also INSERT, UPDATE, DELETE, and even DDL queries.

10) @RequestParam : annotation is used to read the form data and bind it automatically to the parameter present in the provided method.

11) @Controller: The @Controller is a class-level annotation. It is a specialization of @Component. It marks a class as a web request handler. It is often used to serve web pages. By default, it returns a string that indicates which route to redirect. It is mostly used with @RequestMapping annotation.

12) @RequestMapping is one of the most common annotation used in Spring Web applications. This annotation maps HTTP requests to handler methods of MVC and REST controllers usually used to bind requestParams to custom objects.
Suppose your REST controller is annotated with @InitBinder , every request is handled within that controller will instantiate Initbinder and WebDatabinder will bind the request params to JavaBean objects

14) @Autowired : Autowiring feature of spring framework enables you to inject the object dependency implicitly. It internally uses setter or constructor injection. Autowiring can't be used to inject primitive and string values.

15) @GetMapping : Annotation for mapping HTTP GET requests onto specific handler methods. Specifically, @GetMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod. GET)

16) @Service : annotation is a specialization of @Component annotation. Spring Service annotation can be applied only to classes. It is used to mark the class as a service provider

21) session management :It is a mechanism used by the Web container to store session information for a particular user. Spring Session consists of the following modules: Spring Session Core - provides core Spring Session functionalities and APIs.

22) JSON ! XML
It is based on JavaScript language. ! It is derived from SGML.
It is a way of representing objects. ! It is a markup language and uses tag structure to represent data items.
It does not provides any support for namespaces. ! It supports namespaces.
It supports array. ! It doesn't supports array.

23) API: API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the

weather on your phone, you're using an API.