

Contents

Tech Mahindra Recruitment Process

1. Eligibility Criteria
2. Interview Process
3. Interview Rounds

Tech Mahindra Technical Interview Questions: Freshers and Experienced

4. State the difference between the following:
5. What is the difference between call by value and call by reference?
6. What do you understand by Function Overloading?
7. Name a few Object Oriented Programming languages.
8. What do you understand about Structured Programming?
9. Define inline functions in C and C++. Also, give an example of an inline function in C.
10. What are Destructors in C++? Write down the syntax for a destructor in C++.
11. What is the "finalize" method used for in Java? Give an example illustrating the use of the "finalize" method.
12. What do you understand about tokens in C or C++?
13. Differentiate between a class and an object.
14. State some of the advantages of a DataBase Management System.
15. What are ACID properties in Database Management Systems?
16. What do you understand by a checkpoint in Database Management Systems?
17. What are transparent Distributed Database Management Systems?
18. Define unary operations in Relational Algebra in context to SQL.

Tech Mahindra Technical Interview Questions: Freshers and Experienced

(.....Continued)

19. Explain Java Virtual Machine or JVM.
20. What are the different types of memory areas allocated by the Java Virtual Machine in Java?
21. Explain classloader in Java and its different types.
22. In Java, what is the default value of the local variables?
23. State a few advantages of Java Packages.
24. Calculate the total interest on a number of different loans.
25. Given an array of positive integers, write code to return the differences between the sums of even and odd values in the given array.
26. Differentiate between Swapping and Paging.
27. What is a Request Processor and a Request Dispatcher in context with Java?
28. What do you understand about Servlet Collaboration?
29. State the different states of a process.
30. Define microkernels in context to Operating Systems.
31. What is reentrancy in multiprogramming time sharing systems?
32. Define spooling in Operating Systems.
33. State the pros of multithreaded programming.

Tech Mahindra Interview Preparation

34. Interview Preparation Tips.

Frequently Asked Questions

35. Are the Tech Mahindra interview rounds easy?

Frequently Asked Questions

(.....Continued)

36. Why does one want to join Tech Mahindra?
37. How to prepare for the Tech Mahindra written test exam?
38. What do you know about Tech Mahindra?
39. What is the salary which freshers get in Tech Mahindra?
40. How long is the Tech Mahindra Recruitment Process?
41. What are your areas of strength and weakness?
42. Why are you suitable for this role?
43. What makes you want to change jobs?
44. What are the recruitment policies of Tech Mahindra?

Let's get Started

Even though Information Technology or IT firms have grown really very fast in the last couple of years, particularly in India, only a few of them are able to offer long term employment to their staff. That is why people prefer to work for well-known companies. Being associated with a brand not only assures you that you are on the right route in your career, but it also helps to secure some amount of job security. People feel Tech Mahindra can help them progress their careers while still allowing them to maintain a healthy work-life balance.

Tech Mahindra is a multinational technology corporation based in India that specializes in [Information Technology](#) (IT) and Business Process Outsourcing (BPO). The company is located in Pune and has its registered office in Mumbai, and it is a subsidiary of the Mahindra Group. Tech Mahindra is the United States \$5.2 billion firms as of April 2020 and employs more than one lakh people in 90 countries presently. In the Fortune India 500 list for 2019, the company was placed at rank five in India's IT firms and ranked forty-seven overall. Tech Mahindra announced the conclusion of a merger with Mahindra Satyam on June 25, 2013. As of April 2020, Tech Mahindra had more than nine hundred active clients.

Tech Mahindra is one of the biggest employers in the world right now. Tech Mahindra is able to attract and develop the best talent by providing holistic long-term jobs based on a digital learning environment that is accessible anytime, anywhere, and on any device. As a new employee, Tech Mahindra is an amazing place to start. It offers a fantastic workplace as well as a welcoming environment with a good ambience conducive to individual and corporate growth.

This article not only explains how to answer relevant Tech Mahindra Interview Questions but also provides a path for both experienced and fresher candidates to obtain their dream job in Tech Mahindra.



Tech Mahindra Recruitment Process

1. Eligibility Criteria

Tech Mahindra's fresher eligibility criteria are identical to those of most other organizations. Before applying for the Tech Mahindra Recruitment Process, one should take a look at the eligibility criteria (both Graduation criteria and Academic Criteria) for the Tech Mahindra recruitment process:

- **Graduation Criteria:** The graduation criteria for the Tech Mahindra Recruitment Process are as follows:-

Criterion for Graduation	Details
Department/Branch of Study	Bachelor's of Engineering (B.E.) / Bachelor's of Technology (B. Tech) in one of the following disciplines: CSE/ECE/IT/EEE/TELECOM/EI. OR M.Sc (Master of Science) in Computer Science & Information Technology.
Mode of Study	Full-Time courses recognised by the Central or State governments of India. (Not the part-time or correspondence courses available.)
Percentage with which Graduation needs to be done	Minimum Sixty Percent (60%)
Backlogs	No Backlogs are active during the Tech Mahindra Recruitment Process.

- **Academic Criteria:** The academic criteria for the Tech Mahindra Recruitment Process are as follows:-

Academic Criteria	Details
Percentage required in 10th and 12th Standard	For working at Tech Mahindra, one should have sixty per cent or more marks in the 10th and 12th (or diploma) standards.
Educational Gap Allowed	After Higher Secondary Exams (12th standard), but not after Secondary School Certificate Exams(10th standard), or between semesters of graduation, a maximum break of one year is permitted.

- **General Criteria (or General Skills demanded):** There are some general skills that are demanded out of a candidate for working at Tech Mahindra. Some of these are as follows:-
 - Excellent Communication and Interpersonal Skills.
 - Good Knowledge in the Technical Domain.
 - Professional Attitude towards work.
 - Great Aptitude Skills.

2. Interview Process

The Tech Mahindra Interview Process for freshers is almost similar to that of Experienced except for the difficulty of questions asked in the various rounds and the number of rounds (Experienced Folks may be directly interviewed and may be allowed to skip the first few rounds as decided by the Recruiter).

3. Interview Rounds

Round 1(Written Test): The first round in the Tech Mahindra Recruitment Process is a written test to evaluate a candidate's Aptitude and Technical skills. It consists of the following three sections:

- Aptitude
- Essay Writing in English
- Technical Test

The first round of the Tech Mahindra Recruitment Process is an elimination round and only a certain fixed number (as decided by the company) of candidates proceed to round two.

Round 2(Psychometric Test): (optional)

In the Tech Mahindra interview process, a Psychometric Test is taken for the candidates who clear Round one. This round is a personality psychometric that is used to assess work behaviour. This round is a non-elimination round in which there is a written test (now online) with around eighty questions for the candidates to solve in about 20/30 minutes.

Round 3(Technical Written Test)

In the Tech Mahindra selection process, Round 3 is again a written test intended to assess a candidate's technical abilities and aptitude. This round consists of the following sections, which must be completed in under two hours:

- Analytical and Numerical Ability
- Software Testing Methodologies
- Data Structures and Algorithms
- Pseudo Code
- Linux
- Software Testing Concepts
- Database Query Languages like SQL (Structured Query Language)
- Fluid Intelligence
- Hands-on Programming

This round is also eliminative in nature.

Note: Above all rounds may or may not be held for experienced candidates based on company requirements and job positions.

Round4(Technical interviews):

Questions about any aspect of the candidate's studies can be expected in this round. To pass this round, one must have a very good understanding of Computer Science fundamentals. A candidate should be familiar with data structures, algorithms, database management systems, operating systems, networking, object-oriented programming ideas, and a programming language of their choice like C++, Java, Python, etc. Candidates from branches other than computer science should study subjects related to their field. During the interview, Computer Science candidates will be put to the test in terms of code. The panel will also ask questions from the resume, so be sure to know what you've put in your CV (Curriculum Vitae). This round might also include puzzle related questions.

Round 5(HR Round):

If a candidate has cleared the technical interviews, a final HR (Human Resource) Round takes place in which behavioural questions are asked to the candidate to test whether he or she is a cultural fit for the company or not. Most candidates assume that the HR interview is simple, but keep in mind that even if you have cleared all other hurdles, a poor HR interview can endanger your chances of getting the job. Maintaining a nice and confident demeanour is the goal. Remember to smile throughout interviews because they can be long and tedious. The hiring process comes to a close at this point. Adaptability is demanded of candidates, although this stage allows for negotiation. Some of the questions which are asked during this round are as follows:

- Please tell me a little bit about yourself.
- Why do you wish to work in the field of information technology?
- What makes you want to work at Tech Mahindra?
- Are you open to relocating to various locations in India?
- What do you think your pay will be? Answering this question is challenging. It is a question that most experienced employees are asked. You might inquire about the company's standard employee rise.

Tech Mahindra Technical Interview Questions: Freshers and Experienced

4. State the difference between the following:

- `#include <file>`
- `#include "file"`

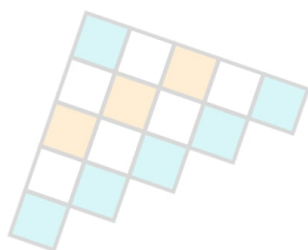
The main difference between the two is in the search location for the included file of the preprocessor. The preprocessor looks for `#include "file"` in the same directory as the directive's file. Normally, this approach is used to incorporate programmatically created header files. On the other hand, the preprocessor searches for `#include <file>` usually in search directories pre-designated by the compiler or the IDE (Integrated Development Environment) and not necessarily in the same directory as the directive's file. In most cases, this approach is used to incorporate standard library header files.

5. What is the difference between call by value and call by reference?

Let us first take a look at the definitions of Call by Value and Call by Reference before taking a look at the differences between the two:

- **Call By Value:** In this way of parameter passing, the values of actual parameters are copied to the formal parameters of the function, and the two types of parameters are stored in separate memory regions. As a result, any changes made inside functions are not reflected in the caller's actual arguments.
- **Call by Reference:** In Call by Reference, both the actual and formal parameters refer to the identical locations and because of it, any changes performed within the function are reflected in the caller's actual parameters.

Let us now take a look at the differences between Call by Value and Call by Reference in detail:



CALL BY VALUE	CALL BY REFERENCE
In Call by Value, the value of each variable in the calling function is transferred into the called function's corresponding dummy variables.	In Call by Reference, the addresses of the calling function's actual variables are copied into the called function's dummy variables.
In Call by Value, changes to the dummy variables in the called function have no influence on the values of the actual variables in the calling function.	Since addresses of the actual variables of the calling function are being passed in the called function, any changes made to the variables in the called function gets reflected in the variables of the calling function also.
We can't change the values of actual variables through function calls in call by values.	In call by reference, we can change the actual values of variables through function calls.

An example each to illustrate the difference between Call by Value and Call by Reference in C is given below:

Call by Value Example:

```
// Illustrating call by value in C
#include <stdio.h>
// Prototype of the function "addNumbers"
void addNumbers(int a, int b);
// The main function of the C program
int main()
{
    int x = 50, y = 30;
    // Passing x and y by Values
    addNumbers(x, y);
    printf("x = %d y = %d\n", x, y);
    return 0;
}
// Function to add two Numbers and stores the result in the first number passed
void addNumbers(int a, int b)
{
    a = a + b;
    printf("a = %d b = %d\n", a, b);
}
```

Output:

```
a = 80 b = 30
a = 50 b = 30
```

NOTE: The values of the parameters passed did not change in Call By Value, that is, x remained as 50 and y remained as 30.

Call by Reference Example:

```
// Illustrating call by value in C
#include <stdio.h>
// Prototype of the function "addNumbers"
void addNumbers(int *a, int *b);
// The main function of the C program
int main()
{
    int x = 50, y = 30;
    // Passing x and y by Values
    addNumbers(&x, &y);
    printf("x = %d y = %d\n", x, y);
    return 0;
}
// Function to add two Numbers and stores the result in the first number passed
void addNumbers(int *a, int *b)
{
    *a = *a + *b;
    printf("a = %d b = %d\n", a, b);
}
```

Output:

```
a = 80 b = 30
a = 80 b = 30
```

NOTE: The values of the parameters passed changed in Call By Reference, that is, x became 80 and y became 30.

6. What do you understand by Function Overloading?

Object-Oriented programming has a feature called function overloading, which allows two or more functions to have the same name but distinct parameters.

Function Overloading occurs when a function name is overloaded with several jobs. The "Function" name should be the same in Function Overloading, but the arguments should be different. Polymorphism is a feature of C++ that can be used to overload functions. An example of Function Overloading in C++ is given below:

```
#include <bits/stdc++.h>
using namespace std;
void foo(int n) {
    cout << " Integer Value:  " << n << endl;
}
void foo(double n) {
    cout << " Decimal Value " << n << endl;
}
void foo(char n) {
    cout << " Character Value" << nc << endl;
}
int main() {
    foo(40);
    foo(452.144);
    foo("A");
    return 0;
}
```

7. Name a few Object Oriented Programming languages.

A few Object Oriented Programming languages are as follows:

- C++
- Java
- Python
- PHP
- JavaScript.

8. What do you understand about Structured Programming?

Structured Programming is a programming paradigm that involves a totally structured control flow. Structure refers to a block, such as (if/then/else), (while and for), block structures, and subroutines, that contains a set of rules and has a defined control flow. Structured programming is used in nearly all programming paradigms, including the OOPs model.

9. Define inline functions in C and C ++. Also, give an example of an inline function in C.

An inline function in C or C++ is one that is declared with the keyword "inline". It serves two purposes:

- It is a compiler directive that requests (it is not mandatory for the compiler to comply with the request of the inline function) that the compiler replaces the body of the function inline by performing inline expansion, that is, putting the function code at the location of each function call, hence reducing the overhead of a function call. It is similar to the register storage class specifier in this regard, which also gives an optimization indication. Inline functions are usually used for frequent calls to the same function.
- The second goal of the keyword "inline" is to alter link behaviour. This is required by the C/C++ separate compilation and linkage model, notably because the function's definition (body) must be replicated in all translation units where it is used to allow inlining during compilation, which creates a collision during linking if the function has external linkage (it violates uniqueness of external symbols). This is handled differently in C and C++ (as well as dialects like GNU C and Visual C++).

An example of an inline function in C is given below:

```
inline void addNumbers(int a, int b)
{
    int ans = a + b;
    printf("Sum of the two numbers is: %d", ans);
}
```

Let us say that the above inline function is called somewhere in the main function of the C program as:

```
int x = 10;
int y = 20;
addNumbers(x, y);
```

Here, the "addNumbers(x, y);" function call will be replaced by the following piece of code in the main function itself by the compiler:


```
int x = 10;
int y = 20;
int ans = x + y;
printf("Sum of the two numbers is: %d", ans);
```

10. What are Destructors in C++? Write down the syntax for a destructor in C++.

Destructors in C++ are instance member functions that are automatically called whenever an object is destroyed. In other words, a destructor is the last function called before an object is destroyed. It is worth noting that if an object was formed with the "new" keyword or if the constructor used the "new" keyword to allocate memory from the heap memory or the free store, the destructor should free the memory with the "delete" keyword.

Destructors are usually used to deallocate memory and do other cleanups for a class object and its class members when the object is destroyed and are called for a class object when that object passes out of scope or is explicitly deleted.

The syntax for a destructor in C++ is given below:

```
~constructor-name();
```

So, for example, if the name of the class is "Car", the destructor of the class would be as follows (the name of the constructor would be "Car"):

```
~Car();
```

11. What is the "finalize" method used for in Java? Give an example illustrating the use of the "finalize" method.

Before an object is destroyed, the "finalize" method or function is used to conduct cleanup operations on unmanaged resources owned by the current object. Because this method is protected, it can only be accessed through this class or a derived class.

The syntax of the finalize method is given below:

```
protected void finalize throws Throwable{}
```

An example illustrating the use of the finalize method is given below:

```
public class FinalizeMethodExample {
    public static void main(String[] args)
    {
        FinalizeMethodExample o = new FinalizeMethodExample ();
        System.out.println(o.hashCode());
        o = null;
        // a call to the garbage collector of Java
        System.gc();
        System.out.println("The end of garbage collection!");
    }
    @Override
    protected void finalize()
    {
        System.out.println("The finalize method is being called now!");
    }
}
```

The output of the above program will be as follows:

```
3202429534
The end of garbage collection!
The finalize method is being called now!
```

12. What do you understand about tokens in C or C++?

A token in C or C ++ is the smallest element of a program that the compiler can understand. Tokens can be of the following types (given along with examples):

- Keywords - "new", "int", "class", etc.
- Strings - "Ritik", "Vaibhav", "Muskan", etc.
- Identifiers - "i", "j", "abc", "temp", etc.
- Special Symbols - "\n", "\r", "\t", etc.
- Constants - 5, 10, 403, etc.
- Operators - +, -, *, etc.

13. Differentiate between a class and an object.

Let us first look at the definitions of a class and an object before looking at the differences between the two:

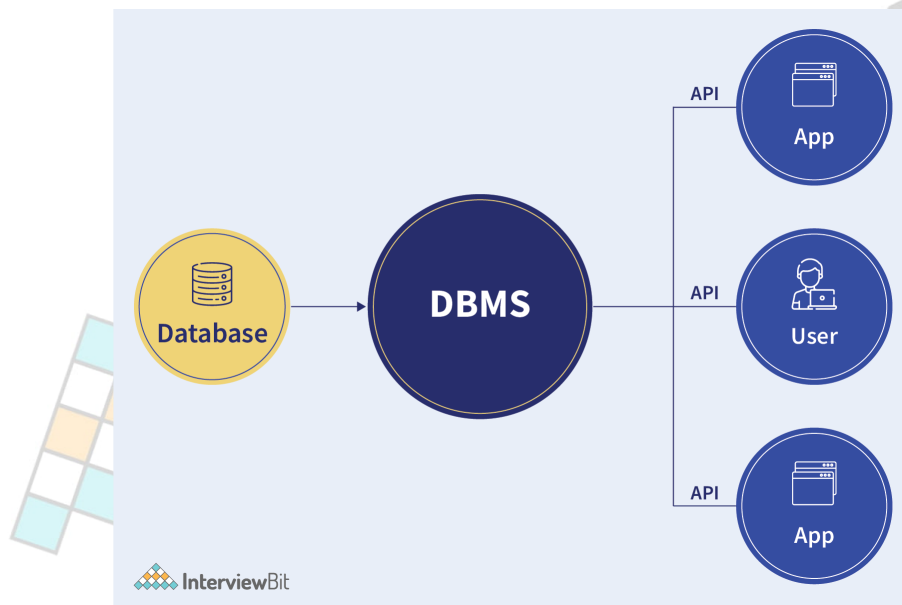
- **Class:** A class is the fundamental building unit of Object-Oriented Programming. It is a user-defined data type with its own set of data members and member functions which can be accessed and used by establishing a class instance (or an object). It is the blueprint of any item. For instance, consider the class "Accountant". There may be a lot of accounts with different names and categories, but they will all have some similar qualities, such as balances, account holder names, and so on. The account is the class, and the amount, as well as the account holder's name, are the properties.
- **Object:** A class's instance is an object. Objects can be used to access all of the class's data members and member functions. When a class is defined, no memory is allocated; nevertheless, memory is allocated when it is instantiated (that is when an object is formed). Consider the objects in the Account class: Savings account, Current account, and so on.

Now that we understand what classes and objects are, let us take a look at the differences between a class and an object:

CLASS	OBJECT
Class is a blueprint or an object factory.	Objects are instances of classes.
On the creation of a class, no memory is allocated as such. Due to not being available in the memory, classes cannot be manipulated.	On the creation of an object, memory is allocated for the manipulation of objects.
It is a logical entity.	Objects are physical entities.
There are no values in the class that can be linked to the field.	Each object has its own set of values that it is associated with.
<pre>class <nameOfClass> { }; </pre>	<pre>// Class Declaration class Account{ public: void foo(){ cout << "This is an Account" << endl; } }; // Main function of the program int main(){ Account newAccount; // Creation of object newAccount.foo(); //Calling the object method } </pre>

14. State some of the advantages of a DataBase Management System.

Some of the advantages of a DataBase Management System are as follows:



- It helps in controlling redundancy in the database.
- Unauthorized access is restricted by it.
- Multiple user interfaces are available.
- Backup and recovery services are available.
- Integrity limitations are enforced.
- Ensure that data is consistent.
- Easily accessible.
- Due to the usage of queries, data extraction and processing are simple.

15. What are ACID properties in Database Management Systems?

The term "ACID" is an acronym for Atomicity, Consistency, Isolation and Durability. A transaction is a logical unit of work that accesses and, in certain cases, updates the contents of a database. Read and write operations are used by transactions to access data. Certain properties are followed before and after a transaction in order to preserve consistency in a database. ACID characteristics are what they are called. Let us take a look at each of these characteristics in detail:

- **Atomicity:** Atomicity means that either the complete transaction occurs at once or it does not occur at all. There is no middle ground, which means that transactions do not take place in stages. Each transaction is treated as a single entity that is either completed or not conducted at all. It entails the following two procedures.
 - Abort: If a transaction aborts, any database modifications are lost.
 - Commit: When a transaction commits, the changes it contains become visible.

The 'All or nothing rule' is another name for atomicity.

Consider the following transaction T, which consists of two transactions: T1 and T2. 100 dollars transferred from account X to account Y.

Before: X: 500	Y: 200
Transaction T	
T1	T2
Read (X) $X := X - 100$ Write (X)	Read (X) $Y := Y + 100$ Write (Y)
After: X: 400	Y: 300

If a transaction fails after T1 but before T2 (for example, after write(X) but before write(Y)), the amount deducted from X but not added to Y is deducted. As a result, the database is in an inconsistent condition. As a result, the transaction must be completed in its entirety to guarantee that the database state is valid.

- **Consistency:** This means that integrity constraints must be maintained before and after the transaction to ensure that the database is consistent. It refers to a database's correctness. The entire amount before and after the transaction must be maintained, as shown in the example above.

Before T, the total is $500 + 200 = 700$.

After T, the total is $400 + 300 = 700$.

As a result, the database is consistent. When T1 succeeds but T2 fails, there is inconsistency. As a result, T is not complete.

- **Isolation:** This attribute assures that several transactions can take place at the same time without causing database state inconsistencies. Transactions take place in a non-interfering manner. Changes made in one transaction are not visible to other transactions until that transaction's update is written to memory or committed. This feature assures that concurrently executing transactions results in a state that is identical to the one attained if they were executed sequentially in some order.

Let us take an example to understand Isolation. Let $X = 500$ and $Y = 500$ and let there be two transactions T and T".

T	T''
Read (X) X: = X*100 Write (X) Read (Y) Y: = Y-50 Write (Y)	Read (X) Read (Y) Z: = X+Y Write (Z)

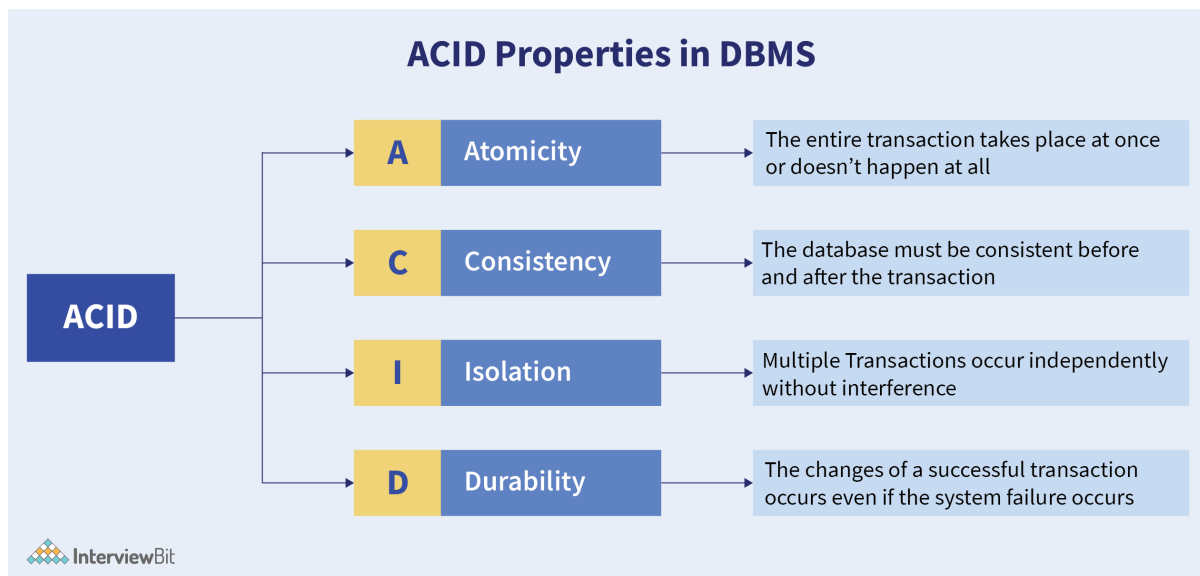
Assume T has been running until Read (Y), at which point T'' begins. As a result of this interleaving, T'' reads the right value of X but the wrong value of Y, and the sum computed by T'': $(X+Y = 50,000+500=50,500)$ is inconsistent with the sum at the end of the transaction:

$T = 50,000 + 450 = 50,450$ ($X+Y = 50,000 + 450 = 50,450$).

Due to the loss of 50 units, this causes database discrepancy. As a result, transactions must be performed in isolation, and changes should only be seen after they have been written to the main memory.

- **Durability:** This attribute ensures that after the transaction has been completed, the database updates and modifications are saved and written to disc and that they survive even if the system fails. These changes are now saved in non-volatile memory and are permanent. The transaction's effects are never lost as a result of this.

The **ACID properties**, taken together, provide a mechanism for ensuring the correctness and consistency of a database in such a way that each transaction is a group of operations that acts as a single unit, produces consistent results, is isolated from other operations, and the updates it makes are durably stored.



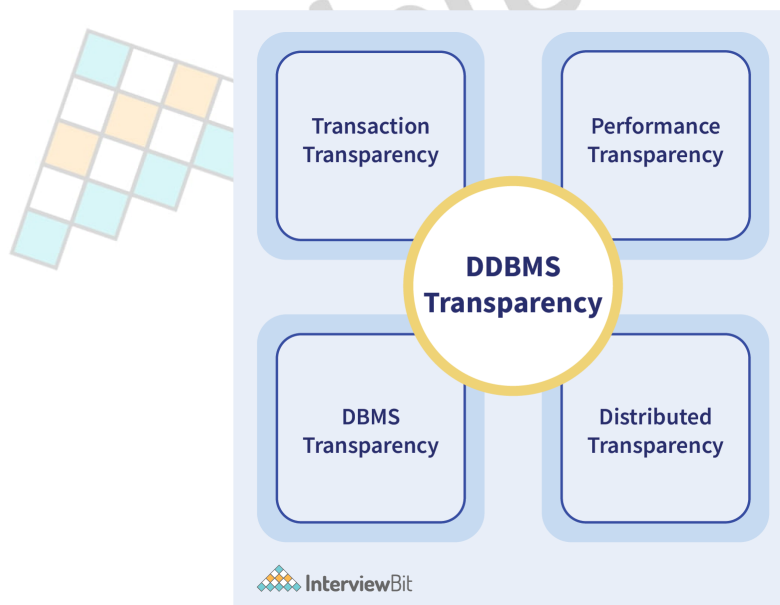
16. What do you understand by a checkpoint in Database Management Systems?

The Checkpoint in Database Management Systems (DBMS) is a technique that removes all previous logs from the system and stores them permanently on the storage drive. Preserving the log of each transaction and maintaining shadow pages are two methods that can assist the DBMS in recovering and maintaining the ACID properties. When it comes to a log-based recovery system, checkpoints are necessary. Checkpoints are the minimal points from which the database engine can recover after a crash as a specified minimal point from which the transaction log record can be utilised to recover all committed data up to the moment of the crash.

17. What are transparent Distributed Database Management Systems?

The transparent Distributed Database Management System is a form of a database management system that conceals its physical structure from users. Physical structure, also known as physical storage structure, refers to the DDBMS's memory manager and explains how data is saved on a disc. Because of this, there is less abstraction. There are four types of transparencies in Distributed Database Management Systems (or DDBMS):

- Transaction transparency
- Performance transparency
- DBMS transparency
- Distribution transparency.



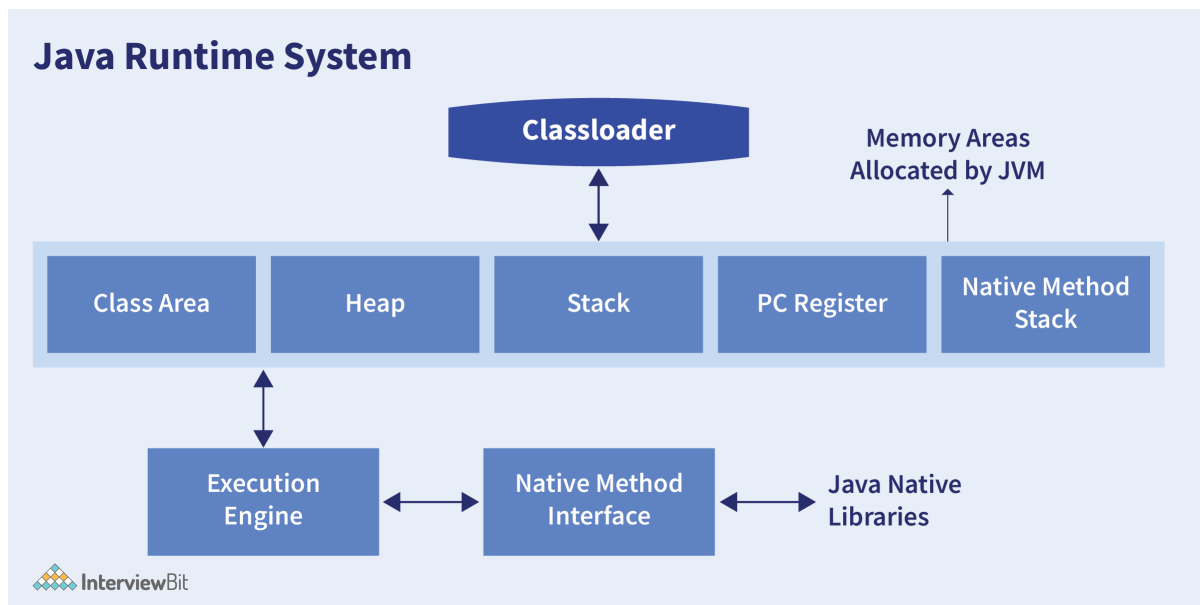
18. Define unary operations in Relational Algebra in context to SQL.

Single operand operations are known as unary operations in Relational Algebra. In relational algebra, the operations PROJECTION and SELECTION and RENAME are unary operations:

- **SELECT:** The SELECT operation is used to select a subset of tuples from a relation that meets a set of criteria. The SELECT operation can be thought of as a filter that keeps only those tuples that meet a set of criteria. Alternatively, the SELECT operation can be used to limit the tuples in relation to those that meet the criterion. The SELECT action can also be thought of as a horizontal division of the relation into two sets of tuples: those that fulfil the condition and are selected, and those that do not and are rejected.
- **PROJECT:** If we consider a relation to be a table, the SELECT operation selects some of the rows while discarding others. The PROJECT operation, on the other hand, picks a subset of the table's columns while discarding the rest. If we're just interested in a few of a relation's attributes, we may use the PROJECT operation to project the relation over just those. As a result, the PROJECT operation's result can be seen as a vertical division of the relation into two relations: one with the required columns (attributes) and the operation's result, and the other with the rejected columns.
- **RENAME:** Breaking a difficult chain of operations and renaming it as a relation with new names is sometimes straightforward and appropriate. There are numerous reasons to rename a relation, including:
 - It is possible that we will wish to save the outcome of a relational algebra expression as a relation so we may use it later.
 - We may want to join relation to itself.

19. Explain Java Virtual Machine or JVM.

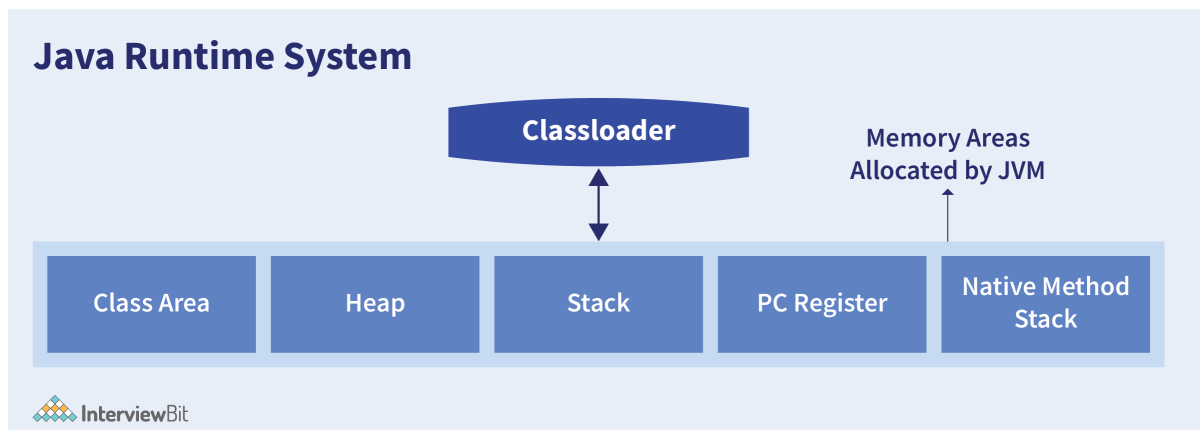
A virtual machine that allows a computer to run a Java programme is known as a Java Virtual Machine. The JVM functions as a run-time engine that calls the main method in Java programming. The Java Virtual Machine (JVM) is a specification that must be implemented in a computer system. JVM compiles Java code into Bytecode, which is machine agnostic and similar to native code. The image below shows the architecture of JVM:



20. What are the different types of memory areas allocated by the Java Virtual Machine in Java?

The different types of memory areas allocated by the Java Virtual Machine are as follows:

- **Class(Method) Area:** The Class(Method) Area maintains per-class structures such as the runtime constant pool, fields, method data, and method code.
- **Stack:** The Java Stack is where frames are stored. It manages local variables and partial results, as well as invoking and returning methods. Each thread has its own JVM stack, which is built concurrently with the thread. Each time a method is called, a new frame is created. When a frame's method invocation is finished, it is destroyed.
- **Program Counter (PC) Register:** The address of the Java virtual machine instruction presently being executed is stored in the PC (programme counter) register.
- **Heap:** This is the runtime data location where the objects' memory is allocated.
- **Native Method Stack:** Each and every native method used in the application is present in it.



21. Explain classloader in Java and its different types.

The Classloader subsystem of the Java Virtual Machine is responsible for loading class files. The classloader loads the java programme first whenever we execute it. The three built-in classloaders of Java are as follows:

- **ClassLoader in Bootstrap:** This is the superclass of Extension classloader, which is the initial classloader. It loads the rt.jar file, which includes all Java Standard Edition class files such as java.lang package classes, java.net package classes, java.util package classes, java.io package classes, java.sql package classes, and so on.
- **Extension ClassLoader:** This is the parent classloader of the System classloader and the child classloader of Bootstrap. The jar files in the `$JAVA_HOME/jre/lib/ext` directory are loaded.
- **System or Application ClassLoader:** Extension classloader's child classloader is System/Application ClassLoader. The class files are loaded from the classpath. The classpath is set to the current directory by default. The "-cp" or "-classpath" switches can be used to alter the classpath. Application classloader is another name for it.

22. In Java, what is the default value of the local variables?

Local variables, primitives, and object references are not initialised to any default value in Java. A code snippet to illustrate the same is given below:

```
public class LocalVariablesExample{
    public void foo() {
        int localVariable;
        localVariable = localVariable * 10;
        System.out.println("Local Variable value is : " + localVariable);
    }
    public static void main(String args[]) {
        LocalVariablesExample obj = new LocalVariablesExample();
        obj.foo();
    }
}
```

The following error is generated while compiling this code because the local variable in the function foo is not initialised:

```
LocalVariablesExample.java:4:variable number might not have been initialized
    localVariable = localVariable * 10;
                   ^
1 error
```

23. State a few advantages of Java Packages.

A few advantages of Java Packages are as follows:

- Name collisions are avoided by using packages.
- It is a lot easier to find the classes that are linked.
- The packages make it easier to manage access.
- We can also have hidden classes that are not accessible outside of the package but are used by it.

24. Calculate the total interest on a number of different loans.

However, there is no interest until the sum of 1000 is reached, but there is a 20% interest rate for the remaining quantities in the array. In this scenario, we will write a function "totalInterest" that takes two arguments: the first is the number of amounts in the array, and the second is the array of amounts. For example, if the first input is 4, and the amounts are 1000, 2000, 3000 and 4000. The total interest would be = 20% of 1000 + 20% of 2000 + 20% of 3000 + 20% of 4000 = 2000.

The C++ code for the above-mentioned problem is as follows:

```
#include<bits/stdc++.h>
using namespace std;
double totalInterest(int totalAmounts, vector<int> a)
{
    double totalInterestValue = 0;
    for(int i = 0; i < totalAmounts; i++)
    {
        if(arr[i] < 1000)continue;
        totalInterestValue = totalInterestValue + (1.0*(arr[i]-1000)*0.2);
    }
    return totalInterestValue;
}
int main()
{
    int totalAmounts;
    cin >> totalAmounts;
    vector<int> a(totalAmounts);
    for(int i = 0; i < totalAmounts; i++)cin >> a[i];
    cout << totalInterest(totalAmounts, a) << endl;
    return 0;
}
```

In the code given above, we first declare a decimal variable "totalInterestValue" which would store the total interest yielded in our question. After that, we keep on incrementing its value by adding the total interest generated for each amount given if the value of the amount is greater than or equal to 1000. The interest is calculated only for amounts greater than 1000 and at a rate of 20%.

25. Given an array of positive integers, write code to return the differences between the sums of even and odd values in the given array.

In this scenario, we would have to write a function "evenOddDiff" which takes two inputs as parameters: the number of positive integers in the array and the array of positive integers. For example, if the initial input is 3 numbers and the numbers are 5, 2 and 4, the difference between the sums of even and odd numbers is $(4+2) - (5) = 6 - 1 = 5$

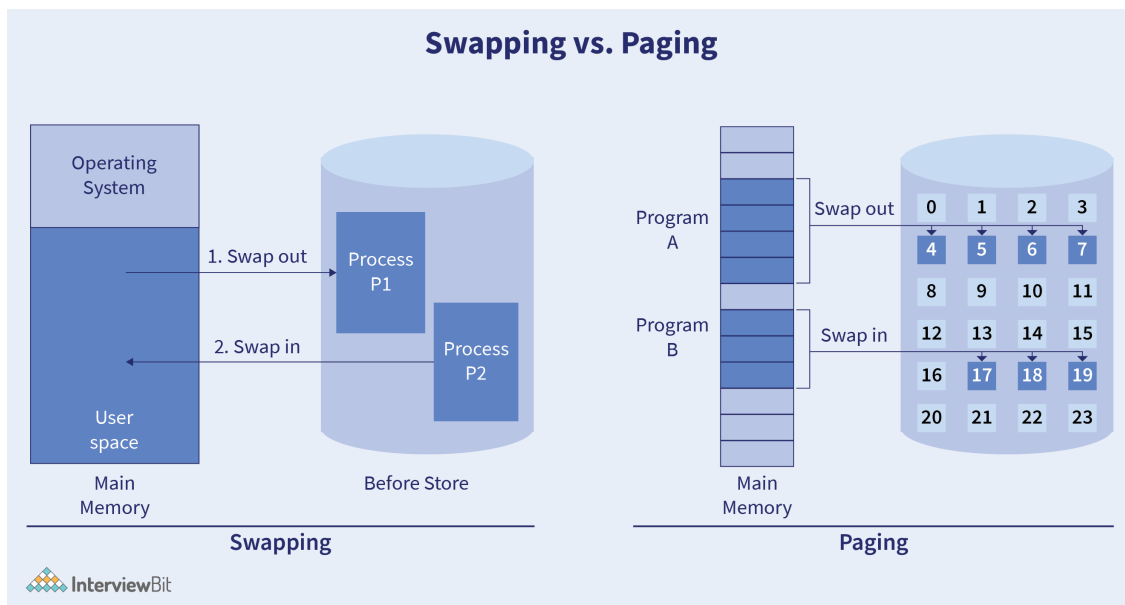
The C++ code for the above-mentioned problem is given below:

```
#include<bits/stdc++.h>
using namespace std;
int evenOddDiff(int n, vector<int> &a)
{
    int oddSum = 0, evenSum = 0;
    for(int i: a)
    {
        if(i & 1)
            oddSum += i;
        else
            evenSum += i;
    }
    return evenSum - oddSum;
}
int main()
{
    int n;
    cin>>n;
    vector<int> a(n);
    for(int j = 0; j < n; j ++){cin >> a[j];}
    cout << evenOddDiff(n, a) << endl;
    return 0;
}
```

In the above-given code, we simply iterate over the elements of the list (given to us in the form of a C++ vector) and check whether each element of the list is an odd number or an even number using the condition "if(i && 1)". We also maintain two variables "oddSum" and "evenSum" which store the sum of odd numbers in the list and the sum of the even numbers in the list respectively. In the end, we simply return the difference between the values of the variables evenSum and oddSum as expected in the question.

26. Differentiate between Swapping and Paging.

The key differences between Swapping and Paging are as follows:



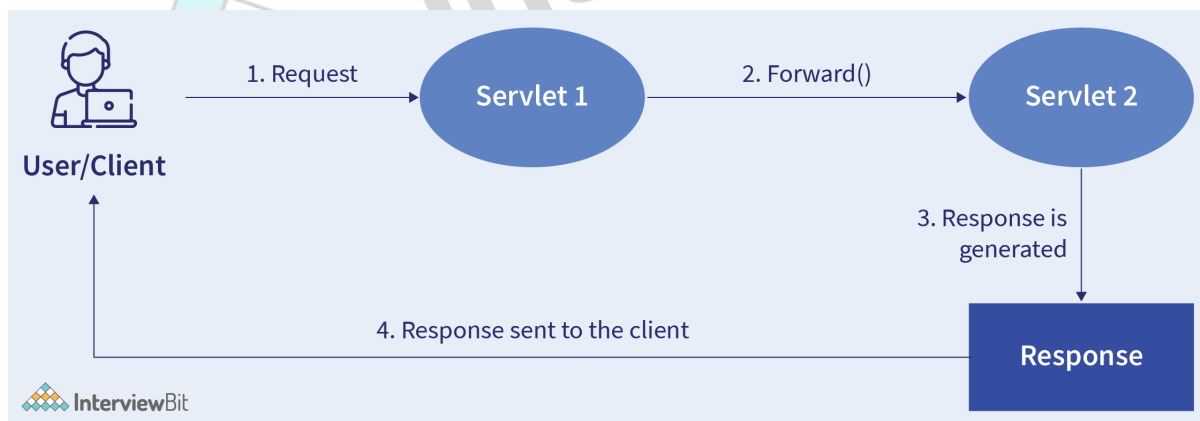
SWAPPING	PAGING
When the entire process is shifted to secondary memory, it is referred to as swapping.	Paging takes place in the event of some part of a process being transferred to secondary memory.
Swapping involves the temporary transfer of a process from the main memory to secondary memory.	In Paging, the contiguous block of memory is made non-contiguous but of fixed size called frame or pages.
No memory management is required for swapping.	Paging requires non-contiguous Memory Management.
Swapping provides the direction regarding the solution in it.	Paging does not give any direction regarding the solution in it.

27. What is a Request Processor and a Request Dispatcher in context with Java?

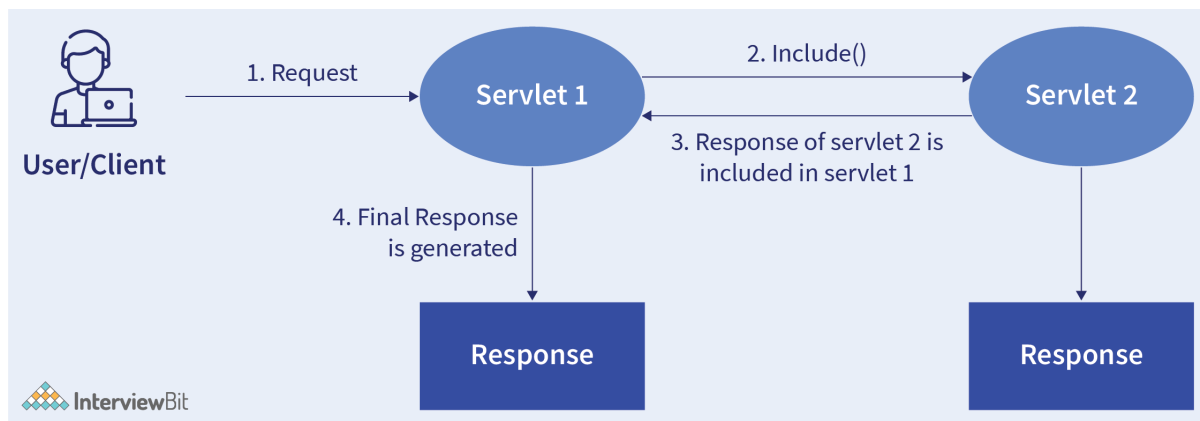
Request Processor: Request processor is a class given by the struts framework that is responsible for handling requests and responses. If we want to change our controller, we can do so in this class. Since version 7.16, the Request processor has been able to perform asynchronous requests in a dedicated thread pool. The majority of RequestProcessor use cases begin with building your own RequestProcessor instance (which by itself is quite lightweight).

Request Dispatcher: Request Dispatcher is an interface that allows requests to be dispatched from one page to another within an application (page may be a servlet file, JSP, etc.). The RequestDispatcher interface offers the following two approaches:

- **public void forward(ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException:** The forward() method is used to redirect a client's request to a different resource (HTML file, servlet, jsp etc). The control is passed to the next resource called when this method is invoked. The include() method, on the other hand, inserts the content of the calling file into the called file. The control remains with the calling resource after calling this method, but the processed output is included in the called resource. The process is depicted in the diagram below:



- **public void include(ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException:** The include() method is used to pass the contents of the caller resource to the called resource. Control remains with the caller resource when this method is called. It merely incorporates the calling resource's processed output into the called one. The diagram below depicts how it works:



28. What do you understand about Servlet Collaboration?

Servlet Collaboration is the process of exchanging information among the servlets of a Java web application. This allows information to be passed from one servlet to another via method invocations. The Servlet API (Application Programming Interface) of Java, which exposes two interfaces, is the primary technique offered by Java to achieve Servlet Collaboration.

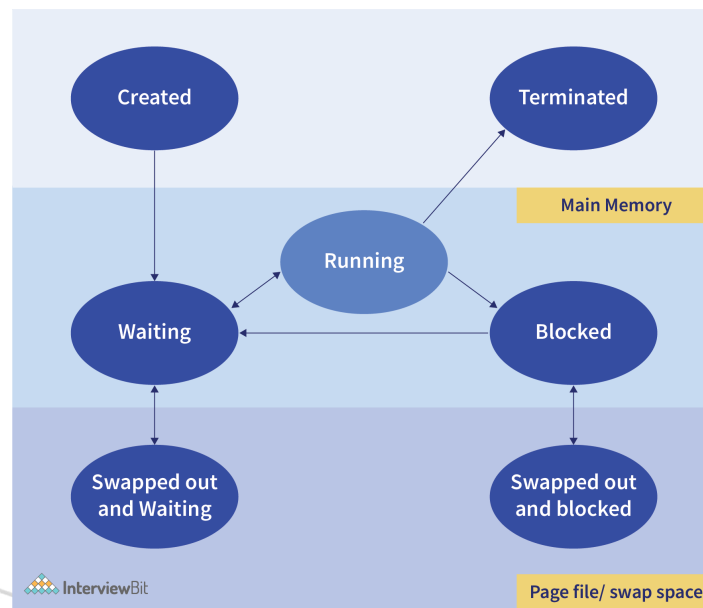
- `javax.servlet.RequestDispatcher`
- `javax.servlet.http.HttpServletResponse`

These two interfaces contain the techniques for fulfilling the goal of information sharing amongst servlets.

29. State the different states of a process.

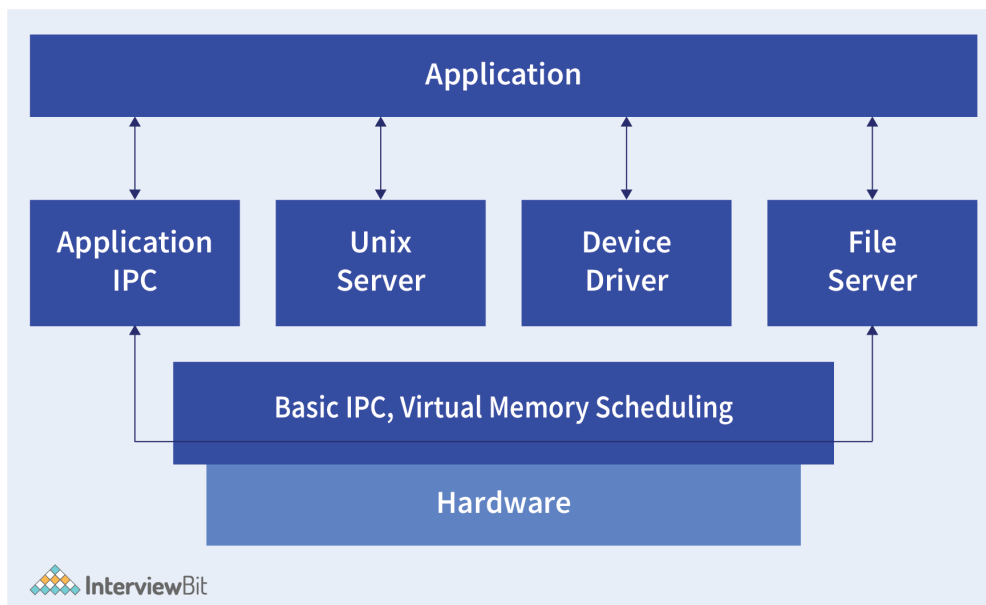
The different states of a process are as follows:

- **Created / New Process:** A New or Created process is a program that will be loaded into the main memory by the operating system.
- **Ready Process:** When a process is created, it immediately goes into the ready state, where it waits for the CPU to be assigned to it. The operating system selects new processes from secondary memory and places them all in the main memory. Ready state processes are processes that are ready for execution and sit in the main memory. Many processes may be active in the ready state.
- **Running Process:** The Operating System will choose one of the processes from the ready state based on the scheduling mechanism. As a result, if our system only has one CPU, the number of running processes at any given time will always be one. If the system has n processors, we can have n processes executing at the same time.
- **Blocked or Waiting Process:** Depending on the scheduling method or the intrinsic behaviour of the process, a process can migrate from the Running state to the Block or Wait states. When a process waits for a specific resource to be assigned or for user input, the Operating System moves it to the block or wait state and assigns the CPU to other processes.
- **Terminated Process:** A process enters the termination state when it has completed its execution. The process's context (Process Control Block) will be removed as well, and the process will be terminated by the operating system.



30. Define microkernels in context to Operating Systems.

A microkernel is one of the kernel's classes. It manages all system resources because it is a kernel. However, user and kernel services are implemented in different address regions in a microkernel. User services are stored in user address space, while kernel services are stored in kernel address space, resulting in a smaller kernel and operating system. It provides the bare minimum of process and memory management services. Message passing is used to establish communication between client programs/applications and services executing in user address space, which slows down the execution microkernel. The Operating System is unaffected because user and kernel services are separated, thus if one fails, the kernel service is unaffected. As a result, it adds to one of the microkernel's benefits. It is easily extensible, in the sense that new services are introduced to the user address space rather than the kernel address space, requiring no changes in the kernel. It is also lightweight, safe, and dependable. Given below is a microkernel's architecture:



Some of the advantages of Microkernel are as follows:

- The small and isolated architecture of this kernel makes it function better.
- Expansion of the system is easier, it is simply added to the system application without disturbing the kernel.

31. What is reentrancy in multiprogramming time sharing systems?

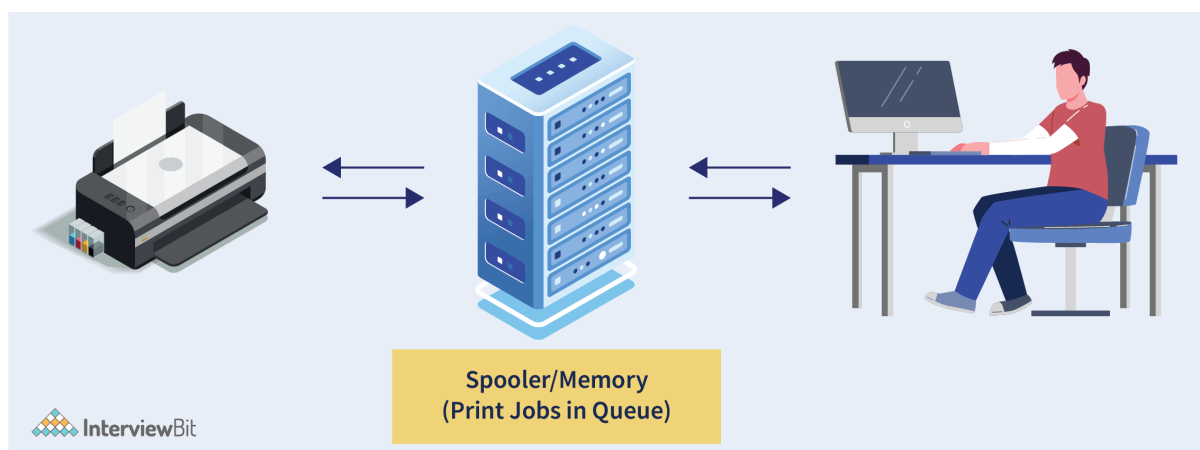
For multiprogramming time-sharing systems, reentrancy is a very effective memory saving strategy. It has the capability of allowing several users to share a single copy of the program at the same time. Two of its main features are as follows:

- The programme code is incapable of self-modification.
- Each user process's local data must be saved separately.

32. Define spooling in Operating Systems.

Spooling is the process of temporarily storing data so that it can be used and processed by a device, program, or system. Data is delivered to and held in memory or other volatile storage until it is requested for execution by a program or computer. SPOOL is an acronym that stands for "Simultaneous Peripheral Operations Online." The spool is usually kept in the computer's physical memory, buffers, or interrupts for Input / Output devices. The FIFO (first-in, first out) method is used to process the spool in ascending order. Spooling is the process of collecting data from many Input / Output tasks and storing it in a buffer. This buffer is a section of memory or hard disc that Input / Output devices can access. An operating system performs the following tasks in a distributed environment:

- Handles data spooling for Input / Output devices, which have various data access rates.
- Maintains the spooling buffer, which serves as a holding area for data while the slower device catches up.
- Because a computer can do Input / Output in parallel order, the spooling process maintains parallel computing. It is now feasible for the computer to read data from a tape, write data to disc, and print data to a tape printer all at the same time.



Printing is the most obvious example of spooling. The papers to be printed are stored in the SPOOL before being added to the printing queue. During this time, several programs can run and utilise the CPU without having to wait for the printer to complete the printing process on each paper individually. Many additional features can be added to the Spooling printing process, such as establishing priorities, receiving notifications when the printing process is complete, and choosing different types of paper to print on based on the user's preferences.

33. State the pros of multithreaded programming.

The pros of multithreaded programming are as follows:

- Multithreaded Programming improves the responsiveness of the system to the users.
- Within the process, there is resource sharing in multithreaded programming.
- It is economical in terms of the resources being used in the project since parallel processing is achieved in a way.
- It utilizes the multiprocessing architecture to its full potential.

Useful Interview Resources

- [Coding Interview Questions](#)
- [Online Compiler](#)
- [Coding Problems](#)

Tech Mahindra Interview Preparation

34. Interview Preparation Tips.

It is strongly recommended that the readers of this article take a look at the following tips to prepare for a Tech Mahindra Interview:

Interview Tips



- Test yourself for a set of [coding challenges](#) to see how quickly you can solve them. This will help you improve your problem-solving skills and help you build your logic structures.
- Share your experiences and prepare well to demonstrate leadership skills, teamwork, professional and academic performance, communication skills, and the ability to overcome obstacles and challenges.
- Understand your strengths and weaknesses, as well as the experiences and talents that make you fit for the job. Consider thinking of an example that will help you stand out from the crowd. The interviewer wants to know what motivates you, why you are interested in the company, and how your past qualifies you for that position.
- It is always good to have an optimistic and friendly attitude. Start the interview on a positive note by introducing yourself in a confident manner.
- The more ready you are for an interview, the more likely you are to get a job. Learn all about interviews, including stages, rounds, and questions. Answers to frequently asked questions during human resource development (HR) and management interviews should be prepared in advance. You can also learn about topics such as company performance, organizational structure, vision, and work-life balance.
- Have a look at the company's interview experience articles. This will give you a good idea of how the actual interview will take place and what you can expect from the interview.
- Consider a mock interview. This will give you an idea of how the actual interview will take place. For such interviews, you can try the [InterviewBit](#) platform. You will be matched with your peers and both will be able to have an interview, which will be in your favour.

Frequently Asked Questions

35. Are the Tech Mahindra interview rounds easy?

Well, the answer to this question depends from person to person and one's level of preparation. In general, it is said that Tech Mahindra interview questions are of easy-medium difficulty and if one is confident enough with his or her preparation, one should be good to go and not face a lot of problems in the Tech Mahindra interview process.

36. Why does one want to join Tech Mahindra?

Tech Mahindra attracts and develops top people by providing holistic long-term positions based on a digital learning environment that is accessible at any time, from any location, and on any device. As a new employee, Tech Mahindra is a fantastic place to start your career. It offers an outstanding work environment as well as a welcoming atmosphere with a positive ambience beneficial to personal and corporate progress. As a result, there is sufficient motivation to work at Tech Mahindra.

37. How to prepare for the Tech Mahindra written test exam?

The Tech Mahindra written test exam mostly consists of one written round on Quantitative Aptitude, Logical questions and English Grammar and another the essay writing round. The main mantra to prepare for these rounds is to practice a lot of questions on the topics mentioned and be confident during the exam. Some of the most important Tech Mahindra essay writing topics which one can prepare for the Tech Mahindra written test exam are as follows:

- The impact of Social Media on the youth of today.
- Stress and its effects on Youth.
- Why are Teenagers of today depressed?

38. What do you know about Tech Mahindra?

You can start with a basic introduction to Tech Mahindra. For instance, you can say that Tech Mahindra, a subsidiary of the Mahindra Group, is a multinational technology corporation based in India that specialises in Information Technology (IT) and Business Process Outsourcing (BPO). After that, you can talk about one or two major projects undertaken by Tech Mahindra which you really liked and how they impacted the world, this will make the interviewer feel that you are also interested in working at Tech Mahindra. You can also mention a few of the principles of the company.

39. What is the salary which freshers get in Tech Mahindra?

The average salary for a Fresher Software Development Engineer at Tech Mahindra is around three and a half lakhs per year (3,50,000 per year) in India.

40. How long is the Tech Mahindra Recruitment Process?

The Tech Mahindra Recruitment Process usually lasts for about two to three months based on the availability of the interviewers and the candidates.

41. What are your areas of strength and weakness?

This is indeed one of the most frequently asked questions in Human Resource interviews. Your answer to this question should not look overconfident. At the same time, dealing with mistakes does not seem unreliable. It is always a good practice, to be honest. Some of the advantages that can be mentioned in the interview are as follows:

- I am a team player who is focused and wants to learn new things every day.
- I have good communication skills.
- My ability to cope with pressure and be calm in stressful situations makes me stand out.
- I have excellent leadership skills which might help me in the long run.

Let us now take a look at some of the shortcomings that can be mentioned in the interview:

- I often have self-doubt when I am new to some technology or environment.
- I am self-critical.
- I have stage fear (fear of speaking in front of a large crowd).

It is definitely important to admit mistakes, but it is also important to explain how you worked to correct them. This will have an impression on the interviewer that you are someone who is constantly trying to improve.

42. Why are you suitable for this role?

When you are asked questions like this, you should mention your strengths and skills that are appropriate for the position. One typical answer to this question can be as follows:

"I am convinced that I am fit for this role for a variety of reasons, but above all because of my commitment to keep going. To succeed in this role, I have all the required skills. I am working on learning for myself. My skills and abilities are in perfect agreement with the requirements of this position. Above all, my communication and leadership skills make me a strong candidate for that position."

43. What makes you want to change jobs?

This is a popular question among seasoned professionals looking for a change. The simplest way to respond to this question is to say that you are leaving your current job to pursue a promotion. Make sure you do not criticise or speak negatively about your current employer.

44. What are the recruitment policies of Tech Mahindra?

The recruitment policies of Tech Mahindra are as follows: