CSE TA Interview Questions

This document contains 1000 important interview questions for a Computer Science Teaching Assistant position, covering a wide range of subjects.

C Programming

1. **What is the difference between `malloc()` and `calloc()`?**
   - `malloc()` allocates a single block of memory, `calloc()` allocates multiple blocks and initializes them to zero.
2. **Explain the concept of pointers in C.**
   - Pointers are variables that store memory addresses of other variables.
3. **What is a `NULL` pointer?**
   - A pointer that does not point to any valid memory location. It's often used to indicate the end of a linked list or an uninitialized pointer.
4. **How do you pass arguments by reference in C?**
   - By passing the address of the variable (using `&`) and using a pointer in the function signature.
5. **What is the purpose of `void` pointer?**
   - A generic pointer that can point to any data type. It must be type-casted before dereferencing.
6. **Differentiate between `const char *p` and `char *const p`.**
   - `const char *p`: pointer to a constant character (data cannot be changed, pointer can).
     `char *const p`: constant pointer to a character (pointer cannot be changed, data can).
7. **What is a `static` variable in C?**
   - A variable that retains its value between multiple function calls and has a scope limited to the file or function where it's declared.
8. **Explain `extern` keyword.**
   - Used to declare a global variable or function that is defined in another file, making it accessible across multiple source files.
9. **What is the use of `register` keyword?**
   - Suggests to the compiler to store the variable in a CPU register for faster access, though the compiler may ignore it.
10. **What is a `volatile` keyword?**

* Indicates that a variable's value can be changed by something outside the program's control (e.g., hardware), preventing the compiler from optimizing away accesses to it.

11. **What is a `struct` in C?**

    * A user-defined data type that groups variables of different data types under a single name.

12. **What is a `union` in C?**

    * A special data type that allows different data types to be stored in the same memory location. Only one member can hold a value at a time.

13. **Difference between `struct` and `union`.**

    * `struct` members occupy separate memory locations; `union` members share the same memory location.

14. **What is `enum` in C?**

    * A set of named integer constants, making code more readable and maintainable.

15. **Explain preprocessor directives.**

    * Instructions to the C preprocessor to perform actions before compilation, such as including files (`#include`) or defining macros (`#define`).

16. **What is a macro? How is it different from a function?**

    * A piece of code in a program that is replaced by the value of the macro. Macros are text substitutions, functions are actual code blocks. Macros can lead to unexpected side effects.

17. **What is the difference between `#include <file.h>` and `#include "file.h"` ?**

    * `<file.h>` searches in standard library directories; `"file.h"` searches in the current directory first, then standard directories.

18. **What is the `sizeof` operator?**

    * Returns the size, in bytes, of a variable or a data type.

19. **What is type casting?**

    * Converting a variable of one data type into another data type.

20. **Explain the use of `break` and `continue` statements.**

    * `break` terminates the loop or switch statement; `continue` skips the rest of the current iteration and proceeds to the next.

21. **What is recursion? Give an example.**

    * A function calling itself directly or indirectly. Example: calculating factorial.

22. **What is a dangling pointer?**

* A pointer that points to a memory location that has been deallocated (freed).

23. **What is a null pointer vs. void pointer?**

    * Null pointer points to nothing, indicating an invalid memory address. Void pointer is a generic pointer that can point to any data type.

24. **How do you handle errors in C?**

    * Using return codes, `errno` global variable, and `perror()` function.

25. **What is the difference between `lvalue` and `rvalue`?**

    * `lvalue` refers to an object that occupies some identifiable location in memory (e.g., a variable). `rvalue` refers to a temporary value that does not persist beyond the expression it is used in.

26. **What is `typedef`?**

    * Used to create an alias for existing data types, making code more readable.

27. **Explain storage classes in C.**

    * `auto`, `register`, `static`, `extern`. They define the scope, lifetime, and storage location of variables.

28. **What is dynamic memory allocation?**

    * Allocating memory during program execution using functions like `malloc()`, `calloc()`, `realloc()`, and `free()`.

29. **What is the difference between `const` and `#define`?**

    * `const` is a variable whose value cannot be changed after initialization (type-checked). `#define` is a preprocessor directive for text substitution (no type-checking).

30. **What is the purpose of `main()` function?**

    * The entry point of every C program.

31. **How do you read a line of text from the console in C?**

    * Using `fgets()` or `scanf()` with appropriate format specifiers.

32. **What is the difference between `gets()` and `fgets()`?**

    * `gets()` is unsafe as it doesn't check buffer overflow; `fgets()` is safer as it takes buffer size as an argument.

33. **What is the use of `goto` statement?**

    * Transfers control to a labeled statement within the same function. Generally discouraged due to making code hard to read and maintain.

34. **What are bitwise operators?**

* Operators that perform operations on individual bits of integer types: `&` (AND), `|` (OR), `^` (XOR), `~` (NOT), `<<` (left shift), `>>` (right shift).

35. **Explain operator precedence and associativity.**

    * Precedence determines the order in which operators are evaluated (e.g., `*` before `+`). Associativity determines the order when operators have the same precedence (e.g., left-to-right).

36. **What is the difference between `++i` and `i++`?**

    * `++i` (pre-increment) increments `i` then uses its new value. `i++` (post-increment) uses `i`'s current value then increments it.

37. **What is a self-referential structure?**

    * A structure that contains a pointer to a structure of the same type. Used to create linked data structures like linked lists, trees, etc.

38. **How do you open and close a file in C?**

    * `fopen()` to open, `fclose()` to close.

39. **What are the different file opening modes?**

    * `r`, `w`, `a`, `r+`, `w+`, `a+`.

40. **What is the difference between text mode and binary mode file operations?**

    * Text mode handles newline character conversions; binary mode reads/writes bytes directly without interpretation.

41. **What is `fprintf()` and `fscanf()`?**

    * `fprintf()` writes formatted output to a file; `fscanf()` reads formatted input from a file.

42. **What is `fseek()`?**

    * Used to set the file position indicator for the stream to a new position.

43. **What is `rewind()`?**

    * Sets the file position indicator to the beginning of the file.

44. **What is `ftell()`?**

    * Returns the current file position of the given stream.

45. **What is `perror()`?**

    * Prints a system error message corresponding to the current value of `errno`.

46. **What is the `assert()` macro?**

    * Used for debugging. If its argument evaluates to false, the program terminates with an error message.

47. **What is the difference between `const` and `volatile`?**

* `const` means the value cannot be changed by the program. `volatile` means the value can be changed by external factors, so the compiler should not optimize away reads.

48. **What is an array of pointers?**

    * An array where each element is a pointer.

49. **What is a pointer to an array?**

    * A pointer that points to an entire array.

50. **What is a function pointer?**

    * A pointer that stores the memory address of a function, allowing functions to be passed as arguments or stored in data structures.

51. **How do you declare a function pointer?**

    * `return_type (*pointer_name)(parameter_list);`

52. **What is the use of `vprintf()` and `vfprintf()`?**

    * Used when a function needs to accept a variable number of arguments and pass them to `printf()` or `fprintf()`.

53. **What is the `restrict` keyword?**

    * A type qualifier that indicates that a pointer is the sole initial means of accessing a data object, allowing for better compiler optimization.

54. **Explain the concept of `endianness`.**

    * The order in which bytes are stored in computer memory (Little-endian: least significant byte first; Big-endian: most significant byte first).

55. **What is a memory leak? How to prevent it?**

    * Occurs when dynamically allocated memory is no longer needed but not freed, leading to resource depletion. Prevent by always `free()`ing allocated memory.

56. **What is a segmentation fault?**

    * An error caused by a program trying to access a memory location that it is not allowed to access, often due to dereferencing a dangling or null pointer.

57. **What is the difference between `char *s = "Hello";` and `char s[] = "Hello";`?**

    * `char *s` creates a pointer to a string literal, which is read-only. `char s[]` creates an array on the stack and copies the string into it, making it modifiable.

58. **What is the purpose of the `static` keyword with functions?**

    * It limits the scope of the function to the file in which it is declared, making it invisible to other files.

59. **How can you create a variable number of arguments in a function?**

* Using the `<stdarg.h>` header and its macros: `va_list`, `va_start`, `va_arg`, `va_end`.

60. **What is the difference between `malloc(0)` and `malloc(1)`?**
    * `malloc(0)` behavior is implementation-defined; it may return a `NULL` pointer or a pointer that can be passed to `free()`. `malloc(1)` allocates one byte of memory.
61. **What is the `_Noreturn` function specifier?**
    * Indicates that a function does not return to its caller (e.g., `exit()`).
62. **How do you check if a number is a power of 2 using bitwise operators?**
    * `n > 0 && (n & (n - 1)) == 0`

C++ Programming

1. **What is the difference between C and C++?**
    - C++ is an extension of C, adding object-oriented programming (OOP) features like classes, objects, inheritance, polymorphism, and abstraction.
2. **Explain OOP concepts in C++.**
    - **Encapsulation:** Bundling data and methods that operate on the data within a single unit (class). Hides internal implementation details.

      Inheritance: A mechanism where one class acquires the properties and behaviors of another class.

      Polymorphism: The ability of an object to take on many forms. Achieved through function overloading, operator overloading, and virtual functions.

      Abstraction: Hiding complex implementation details and showing only essential features of the object.
3. **What is a class and an object?**
    - **Class:** A blueprint or template for creating objects. It defines data members and member functions.

      Object: An instance of a class. It has state (data) and behavior (methods).
4. **What is a constructor?**
    - A special member function of a class that is automatically called when an object of that class is created. Used for initializing objects.
5. **What is a destructor?**
    - A special member function that is automatically called when an object is destroyed. Used for cleaning up resources (e.g., freeing dynamically allocated memory).
6. **What is the difference between `new` / `delete` and `malloc()` / `free()`?**

- `new` / `delete` are C++ operators that call constructors/destructors and handle type safety. `malloc()` / `free()` are C library functions that only allocate/deallocate raw memory.

7. **Explain function overloading.**
   - Defining multiple functions with the same name but different parameters (number, type, or order of arguments) within the same scope.
8. **Explain operator overloading.**
   - Giving special meaning to C++ operators when applied to user-defined data types (objects).
9. **What is a virtual function?**
   - A member function declared in a base class that is re-defined (overridden) in a derived class. It enables runtime polymorphism.
10. **What is an abstract class?**

    * A class that contains at least one pure virtual function. It cannot be instantiated and serves as a base for other classes.
11. **What is a pure virtual function?**

    * A virtual function declared with `= 0` in the base class, making the base class abstract. Derived classes *must* implement it.
12. **What is an interface in C++?**

    * Achieved using abstract classes with only pure virtual functions. It defines a contract that derived classes must fulfill.
13. **What is the `this` pointer?**

    * A pointer that points to the current object for which the member function is called. It is implicitly passed to all non-static member functions.
14. **What is a `friend` function and `friend` class?**

    * A `friend` function/class can access the `private` and `protected` members of another class.
15. **Explain `static` members in C++ (data and functions).**

    * **Static data member:** Shared by all objects of the class; initialized only once.

    Static member function: Can only access static data members and static member functions. Can be called without an object.
16. **What is inheritance? Types of inheritance?**

    * A mechanism where one class (derived) acquires properties of another (base). Types: Single, Multiple, Multilevel, Hierarchical, Hybrid.

17. **What is virtual inheritance?**

    * Used to solve the diamond problem in multiple inheritance, ensuring only one copy of the base class sub-object is inherited.

18. **What is a pure virtual destructor?**

    * A destructor declared as pure virtual. It must have an implementation, unlike pure virtual functions, because derived class destructors are called implicitly.

19. **What is the difference between `public`, `private`, and `protected` access specifiers?**

    * **Public:** Accessible from anywhere.

    Private: Accessible only from within the class itself.

    Protected: Accessible from within the class and by derived classes.

20. **Explain templates in C++.**

    * A feature that allows writing generic programs that work with different data types without rewriting the code for each type. Used for generic classes and functions.

21. **What is `std::vector`?**

    * A dynamic array that can resize itself automatically. Part of the C++ Standard Template Library (STL).

22. **What is `std::map`?**

    * An associative container that stores elements formed by a combination of a key value and a mapped value, ordered by key. Part of STL.

23. **What is `std::set`?**

    * An associative container that stores unique elements in a specific order. Part of STL.

24. **What is RAII (Resource Acquisition Is Initialization)?**

    * A C++ programming technique where resource acquisition is tied to object lifetime. Resources are acquired in the constructor and released in the destructor.

25. **Explain smart pointers (`std::unique_ptr`, `std::shared_ptr`, `std::weak_ptr`).**

    * Objects that behave like pointers but automatically manage the memory they point to, preventing memory leaks.

    `unique_ptr`: Exclusive ownership.

    `shared_ptr`: Shared ownership with reference counting.

    `weak_ptr`: Non-owning reference to an object managed by `shared_ptr`, used to break circular references.

26. **What is move semantics (rvalue references, `std::move`)?**

* A C++11 feature that allows resources (like dynamically allocated memory) to be moved from one object to another rather than copied, improving performance.

27. **What is lambda expression in C++?**

    * An anonymous function object that can be defined and used inline. Useful for short, local functions, especially with algorithms.

28. **What is `const` correctness?**

    * The practice of using the `const` keyword to ensure that objects and member functions do not modify data they are not supposed to, improving code safety and clarity.

29. **What is the difference between `pass by value`, `pass by reference`, and `pass by pointer`?**

    * **By value:** A copy of the argument is passed. Changes inside the function don't affect the original.

    By reference: An alias to the original argument is passed. Changes inside the function affect the original.

    By pointer: The memory address of the argument is passed. Changes through dereferencing the pointer affect the original.

30. **What is an initializer list?**

    * A way to initialize data members of a class in the constructor, especially useful for `const` members, references, and base class constructors.

31. **What is the `explicit` keyword?**

    * Prevents implicit conversions from a single-argument constructor, avoiding unintended type conversions.

32. **What is the `mutable` keyword?**

    * Allows a data member of a `const` object to be modified.

33. **What is the difference between `struct` and `class` in C++?**

    * By default, `struct` members are `public`, and `class` members are `private`. Otherwise, they are functionally similar.

34. **What is a virtual destructor?**

    * Ensures that the correct destructor is called when deleting an object through a base class pointer, preventing resource leaks in polymorphic hierarchies.

35. **What is the purpose of `final` and `override` keywords (C++11)?**

    * **`override`:** Explicitly states that a member function is overriding a virtual function from a base class, helping catch errors.

    `final`: Prevents a virtual function from being overridden in derived classes or prevents a class from being inherited.

36. **Explain `try-catch` blocks for exception handling.**

    * `try` block encloses code that might throw an exception. `catch` block handles the exception if one is thrown.

37. **What is `std::exception`?**

    * The base class for all standard library exceptions.

38. **What is RTTI (Run-Time Type Information)?**

    * A mechanism that allows the type of an object to be determined during program execution. Implemented using `dynamic_cast` and `typeid`.

39. **What is `dynamic_cast`?**

    * Used for safe downcasting in polymorphic class hierarchies. Returns `nullptr` if the cast is invalid.

40. **What is `static_cast`?**

    * Used for conversions between related types, like converting `int` to `float` or base class pointer to derived class pointer (without runtime check).

41. **What is `reinterpret_cast`?**

    * Used for low-level, unsafe conversions between unrelated types, like converting a pointer to an integer or vice-versa.

42. **What is `const_cast`?**

    * Used to add or remove `const` or `volatile` qualifiers from a pointer or reference.

43. **What is a namespace?**

    * A declarative region that provides a scope to the identifiers (names of types, functions, variables) inside it. Used to prevent name collisions.

44. **What is `std::string`?**

    * A class that represents character strings, providing more functionality and safety than C-style character arrays.

45. **What is the difference between `std::string` and `char*`?**

    * `std::string` is an object that manages its own memory, provides many utility functions, and is safer. `char*` is a C-style pointer to a character array, requiring manual memory management and lacking built-in functionalities.

46. **What is a copy constructor?**

    * A special constructor that creates a new object as a copy of an existing object of the same class.

47. **What is the copy assignment operator?**

* An operator (`=`) that assigns the contents of an existing object to another existing object of the same class.

48. **What is the Rule of Three/Five/Zero?**

    * **Rule of Three:** If a class defines a destructor, copy constructor, or copy assignment operator, it should define all three.

    Rule of Five: Extends Rule of Three to include move constructor and move assignment operator (C++11).

    Rule of Zero: If a class doesn't manage any resources, it doesn't need any of these special member functions.

49. **What is SFINAE (Substitution Failure Is Not An Error)?**

    * A C++ template metaprogramming technique where invalid substitutions of template parameters during overload resolution are not errors, but instead cause the invalid overload to be discarded.

50. **Explain `std::thread` and multithreading in C++.**

    * `std::thread` is a C++11 class for creating and managing threads, allowing concurrent execution of code.

51. **What are mutexes and `std::mutex`?**

    * Mutexes (mutual exclusion) are synchronization primitives used to protect shared resources from concurrent access by multiple threads, preventing race conditions. `std::mutex` is the C++ standard library implementation.

52. **What is a `condition variable` (`std::condition_variable`)?**

    * A synchronization primitive that allows threads to wait until a particular condition is met. Often used with mutexes.

53. **What is `std::async` and `std::future`?**

    * **`std::async`:** Launches an asynchronous task, potentially in a new thread.

    `std::future`: Provides a mechanism to retrieve the result of an asynchronous operation.

54. **What is a memory model in C++?**

    * Defines how threads interact with memory and how operations on memory are ordered, crucial for concurrent programming.

55. **What is the difference between `nullptr` and `NULL`?**

    * `nullptr` is a keyword introduced in C++11, specifically typed as `std::nullptr_t`, providing type safety. `NULL` is typically a macro defined as `0` or `(void*)0`, which can lead to ambiguity.

56. **What is a variadic template?**

* A template that can take a variable number of template arguments. Used for functions like `printf` or `std::tuple`.

57. **What is `constexpr`?**

    * A keyword that indicates that a value or function can be evaluated at compile time, leading to performance improvements.

58. **What is a perfect forwarding (`std::forward`)?**

    * A technique that allows a function template to forward its arguments to another function while preserving their original value category (lvalue or rvalue).

59. **What is the purpose of `noexcept`?**

    * A specifier that indicates a function does not throw exceptions. Helps the compiler optimize code and provides better error handling guarantees.

60. **Explain the concept of move constructor and move assignment operator.**

    * Special member functions that enable efficient transfer of resources from a temporary object (rvalue) to a new object, avoiding deep copies.

61. **What is a user-defined literal?**

    * Allows you to define your own literal suffixes to create objects of user-defined types directly from literals.

62. **What is `std::optional`?**

    * A C++17 class template that represents an optional value, i.e., a value that may or may not be present. Useful for functions that might not return a value.

Python Programming

1. **What are the key features of Python?**
   - Interpreted, high-level, dynamically typed, object-oriented, extensive standard library, cross-platform.
2. **What is PEP 8?**
   - Python Enhancement Proposal 8 is the style guide for Python code, providing conventions for writing readable code.
3. **Explain the difference between `list` and `tuple`.**
   - `list` is mutable (can be changed), `tuple` is immutable (cannot be changed). Lists use `[]`, tuples use `()`.
4. **What is a dictionary in Python?**
   - An unordered collection of key-value pairs. Keys must be unique and immutable.

5. **What is a set in Python?**
    - An unordered collection of unique elements. Useful for membership testing and removing duplicates.
6. **Explain mutable vs. immutable data types.**
    - **Mutable:** Can be changed after creation (e.g., lists, dictionaries, sets).

      Immutable: Cannot be changed after creation (e.g., numbers, strings, tuples).
7. **What is the purpose of `__init__` method?**
    - The constructor method in Python classes, automatically called when a new object is created. Used for initializing object attributes.
8. **What is `self` in Python classes?**
    - A reference to the instance of the class. It is the first parameter of any instance method.
9. **Explain inheritance in Python.**
    - A mechanism where a new class (child/derived) inherits properties and behaviors from an existing class (parent/base).
10. **What is polymorphism in Python?**

    * The ability of an object to take on many forms. Achieved through method overriding and duck typing.
11. **What is method overriding?**

    * Redefining a method in a child class that is already defined in its parent class.
12. **What is duck typing?**

    * A concept where the type or the class of an object is less important than the methods it defines. 'If it walks like a duck and it quacks like a duck, then it must be a duck.'
13. **What are decorators in Python?**

    * A function that takes another function as an argument, adds some functionality, and returns the modified function.
14. **What are generators in Python?**

    * Functions that use the `yield` keyword to return an iterator. They produce items one at a time and only when required, making them memory efficient.
15. **What is the difference between `range` and `xrange`?**

    * In Python 2, `range` creates a list, `xrange` creates a generator. In Python 3, `range` is the equivalent of Python 2's `xrange` and there is no `xrange`.
16. **What is the GIL (Global Interpreter Lock)?**

* A mutex that protects access to Python objects, preventing multiple native threads from executing Python bytecodes at the same time. This means only one thread can be in a state of execution.

17. **What are `*args` and `**kwargs`?**

    * Used to pass a variable number of arguments to a function. `*args` for non-keyworded arguments (tuple), `**kwargs` for keyworded arguments (dictionary).

18. **What is a lambda function?**

    * A small anonymous function defined with the `lambda` keyword. Can have any number of arguments but only one expression.

19. **What is the difference between `==` and `is`?**

    * `==` checks for equality of values. `is` checks for identity (if two variables point to the same object in memory).

20. **What is a list comprehension?**

    * A concise way to create lists. `[expression for item in iterable if condition]`.

21. **What is a dictionary comprehension?**

    * A concise way to create dictionaries. `{key_expression: value_expression for item in iterable if condition}`.

22. **What is the `pass` statement?**

    * A null statement that does nothing. It is used as a placeholder where syntax requires a statement but no action is needed.

23. **What are modules and packages in Python?**

    * **Module:** A single Python file containing functions, classes, and variables. **Package:** A collection of modules in a directory with an `__init__.py` file.

24. **How do you handle exceptions in Python?**

    * Using `try`, `except`, `else`, and `finally` blocks.

25. **What is the purpose of the `finally` block?**

    * It is always executed, whether an exception occurred or not. Used for cleanup actions.

26. **What is the difference between `append()` and `extend()` for lists?**

    * `append()` adds its argument as a single element to the end of a list. `extend()` iterates over its argument and adds each element to the list.

27. **What is slicing in Python?**

    * A feature that allows accessing a specific range of elements from a sequence type like a list, tuple, or string. `[start:stop:step]`.

28. **What are magic methods (dunder methods)?**

    * Methods with double underscores at the beginning and end of their names (e.g., `__init__`, `__str__`). They are used to implement operator overloading and other special behaviors.

29. **What is the difference between a shallow copy and a deep copy?**

    * **Shallow copy:** Creates a new object but inserts references into it to the objects found in the original. **Deep copy:** Creates a new object and recursively copies all objects found in the original.

30. **What is the `with` statement?**

    * Used to wrap the execution of a block with methods defined by a context manager. Ensures that resources are properly acquired and released.

31. **What is a context manager?**

    * An object that defines the methods `__enter__` and `__exit__`. Used with the `with` statement.

32. **What is pickling and unpickling?**

    * **Pickling:** The process of converting a Python object into a byte stream. **Unpickling:** The inverse operation, converting a byte stream back into an object.

33. **What is the difference between `.py` and `.pyc` files?**

    * `.py` files contain the source code. `.pyc` files contain the compiled bytecode of the Python code, which is what the Python interpreter runs.

34. **What is a virtual environment?**

    * An isolated Python environment that allows you to manage dependencies for different projects separately.

35. **How does memory management work in Python?**

    * Python uses a private heap to manage memory. Memory allocation and deallocation are handled automatically by the Python memory manager, which uses reference counting and a garbage collector.

36. **What is garbage collection in Python?**

    * A process of automatically freeing memory that is no longer in use. Python's garbage collector uses reference counting and a cyclic garbage collector to detect and clean up circular references.

37. **What is the difference between `__str__` and `__repr__`?**

* `__str__` is for creating a user-friendly, readable representation of an object. `__repr__` is for creating an unambiguous, official representation of an object, which should ideally be a valid Python expression to recreate the object.

38. **What is the `enumerate()` function?**

    * Returns an enumerate object, which yields pairs of index and value for an iterable.

39. **What is the `zip()` function?**

    * Returns an iterator of tuples, where the i-th tuple contains the i-th element from each of the argument sequences or iterables.

40. **What is the difference between `for` loops and `while` loops?**

    * `for` loops are used for iterating over a sequence. `while` loops are used for repeating a block of code as long as a condition is true.

41. **What is the `break`, `continue`, and `else` in loops?**

    * `break` exits the loop. `continue` skips the current iteration. The `else` block is executed if the loop completes without being terminated by a `break` statement.

42. **What is a docstring?**

    * A string literal that appears as the first statement in a module, function, class, or method definition. Used to document the code.

43. **What is monkey patching?**

    * Dynamically modifying a class or module at runtime.

44. **What is the difference between `isinstance()` and `type()`?**

    * `isinstance()` checks if an object is an instance of a class or a subclass of it. `type()` returns the exact type of an object and does not account for inheritance.

45. **What is a metaclass?**

    * A class of a class. It defines how a class behaves. A class is an instance of a metaclass.

46. **What is the purpose of the `__slots__` attribute?**

    * A way to explicitly declare data members of a class, which can save memory by not creating a `__dict__` for each instance.

47. **What is `itertools` module?**

    * A module that provides a collection of tools for working with iterators, inspired by functional programming languages.

48. **What is `collections` module?**

    * A module that provides alternatives to Python's general purpose built-in containers, like `dict`, `list`, `set`, and `tuple`. Examples: `namedtuple`, `deque`, `Counter`.

49. **What is the difference between `os` and `sys` modules?**

    * `os` module provides a way of using operating system dependent functionality. `sys` module provides access to system-specific parameters and functions.

50. **What is the difference between `json` and `pickle`?**

    * `json` is a human-readable text format, language-independent, and can only represent a subset of Python's built-in types. `pickle` is a binary format, Python-specific, and can represent a wider range of Python types.

51. **What is the `asyncio` module?**

    * A library to write concurrent code using the `async` / `await` syntax. Used for building asynchronous I/O-bound applications.

52. **What is an `async` function?**

    * A function defined with `async def`. It returns a coroutine object.

53. **What is an `await` expression?**

    * Used to call an `async` function and wait for it to complete. Can only be used inside an `async` function.

54. **What is a type hint?**

    * A way to statically indicate the type of a variable, function parameter, or return value. Used by static analysis tools to find bugs.

55. **What is the `typing` module?**

    * Provides support for type hints.

56. **What is the difference between `input()` and `raw_input()`?**

    * In Python 2, `raw_input()` returns a string, and `input()` tries to evaluate the input as a Python expression. In Python 3, `raw_input()` is renamed to `input()`, and the old `input()` is gone.

57. **What is the `__pycache__` directory?**

    * A directory where Python stores the `.pyc` files (compiled bytecode) of modules.

58. **What is the `if __name__ == '__main__':` block?**

    * A common idiom in Python scripts. The code inside this block will only be executed when the script is run directly, not when it is imported as a module.

59. **What is a `f-string`?**

    * A formatted string literal, prefixed with `f` or `F`. They are faster and more readable than other string formatting methods.

60. **What is the `walrus` operator `:=`?**

* An assignment expression operator introduced in Python 3.8. It allows you to assign a value to a variable as part of an expression.

61. **What is the difference between `remove()`, `del`, and `pop()` on lists?**

    * `remove()` removes the first matching value. `del` removes an item at a specific index. `pop()` removes and returns an item at a specific index (or the last item if no index is specified).

Java Programming

1. **What are the main principles of OOP?**
    - Encapsulation, Inheritance, Polymorphism, and Abstraction.
2. **What is the JVM?**
    - Java Virtual Machine. It's an abstract machine that provides a runtime environment in which Java bytecode can be executed.
3. **What is the difference between JDK, JRE, and JVM?**
    - **JDK (Java Development Kit):** Contains tools for developing Java applications (includes JRE). **JRE (Java Runtime Environment):** Provides the libraries, JVM, and other components to run Java applications. **JVM (Java Virtual Machine):** An abstract machine that executes Java bytecode.
4. **What is the difference between `==` and `.equals()`?**
    - `==` is an operator that compares object references (memory addresses). `.equals()` is a method that compares the content/values of objects.
5. **What is the difference between `String`, `StringBuilder`, and `StringBuffer`?**
    - **String:** Immutable. **StringBuilder:** Mutable, not thread-safe (faster). **StringBuffer:** Mutable, thread-safe (slower).
6. **What is an interface?**
    - A reference type in Java that is a collection of abstract methods. A class can implement multiple interfaces.
7. **What is an abstract class?**
    - A class that cannot be instantiated and may contain abstract methods (methods without a body). It can have constructors and concrete methods.
8. **Difference between abstract class and interface?**
    - A class can extend only one abstract class but implement multiple interfaces. Abstract classes can have constructors, instance variables, and concrete methods; interfaces cannot (before Java 8).

9. **What is method overloading?**
    - Defining multiple methods with the same name but different parameters in the same class.
10. **What is method overriding?**

    * Providing a specific implementation for a method in a subclass that is already defined in its superclass.
11. **What is a constructor?**

    * A special method used to initialize objects. It is called when an object of a class is created.
12. **What is the `this` keyword?**

    * A reference to the current object.
13. **What is the `super` keyword?**

    * A reference to the immediate parent class object.
14. **What is the `final` keyword?**

    * Can be used with variables (to make them constant), methods (to prevent overriding), and classes (to prevent inheritance).
15. **What is the `static` keyword?**

    * Used for a class-level variable or method that is shared among all objects of the class. Static members can be accessed without creating an object.
16. **What is a static block?**

    * A block of code that is executed once when the class is loaded into memory.
17. **What is the difference between `final`, `finally`, and `finalize`?**

    * **`final`:** A keyword to make entities non-modifiable. **`finally`:** A block in a `try-catch` statement that is always executed. **`finalize`:** A method called by the garbage collector before an object is reclaimed.
18. **What is exception handling in Java?**

    * A mechanism to handle runtime errors using `try`, `catch`, `throw`, `throws`, and `finally`.
19. **What is the difference between checked and unchecked exceptions?**

    * **Checked exceptions:** Checked at compile-time (e.g., `IOException`). Must be handled or declared. **Unchecked exceptions:** Not checked at compile-time (e.g., `RuntimeException`). Handling is optional.
20. **What is the `try-with-resources` statement?**

* A `try` statement that declares one or more resources. The resource is automatically closed at the end of the statement.

21. **What is the Collections Framework?**

    * A unified architecture for representing and manipulating collections. Includes interfaces (e.g., `List`, `Set`, `Map`) and implementations (e.g., `ArrayList`, `HashSet`, `HashMap`).

22. **What is the difference between `List`, `Set`, and `Map`?**

    * **`List`:** Ordered collection, allows duplicates. **`Set`:** Unordered collection (mostly), does not allow duplicates. **`Map`:** Key-value pairs, keys must be unique.

23. **What is the difference between `ArrayList` and `LinkedList`?**

    * **`ArrayList`:** Implemented as a dynamic array. Fast for random access (`get`). **`LinkedList`:** Implemented as a doubly linked list. Fast for adding/removing elements (`add`/`remove`).

24. **What is the difference between `HashMap` and `Hashtable`?**

    * **`HashMap`:** Not synchronized (not thread-safe), allows one null key and multiple null values. **`Hashtable`:** Synchronized (thread-safe), does not allow null keys or values.

25. **What is `fail-fast` vs `fail-safe` iterator?**

    * **`fail-fast`:** Throws a `ConcurrentModificationException` if the collection is modified while iterating. **`fail-safe`:** Works on a clone of the collection, so it doesn't throw an exception if the collection is modified.

26. **What is multithreading?**

    * The ability of a program to execute multiple threads concurrently.

27. **How to create a thread in Java?**

    * By extending the `Thread` class or implementing the `Runnable` interface.

28. **What is the difference between extending `Thread` and implementing `Runnable`?**

    * Implementing `Runnable` is preferred because it allows the class to extend another class (Java doesn't support multiple inheritance of classes).

29. **What is the lifecycle of a thread?**

    * New, Runnable, Running, Blocked/Waiting, Terminated.

30. **What is `synchronized` keyword?**

    * Used to create synchronized blocks or methods, ensuring that only one thread can execute them at a time.

31. **What is a `volatile` variable?**

    * A variable whose value is always read from main memory, not from a thread's local cache. Ensures visibility of changes across threads.

32. **What is the difference between `sleep()` and `wait()`?**

    * `sleep()` is a method of the `Thread` class, it pauses the thread for a specified time without releasing the lock. `wait()` is a method of the `Object` class, it makes the thread release the lock and wait until another thread calls `notify()` or `notifyAll()`.

33. **What is a deadlock?**

    * A situation where two or more threads are blocked forever, waiting for each other.

34. **What is the `ExecutorService`?**

    * A higher-level API for managing threads, part of the `java.util.concurrent` package. It provides thread pools and can execute tasks asynchronously.

35. **What is a `Future`?**

    * Represents the result of an asynchronous computation. It provides methods to check if the computation is complete, to wait for its completion, and to retrieve the result.

36. **What is a lambda expression in Java?**

    * An anonymous function that can be used to create instances of functional interfaces.

37. **What is a functional interface?**

    * An interface with a single abstract method. Can be annotated with `@FunctionalInterface`.

38. **What is the Stream API?**

    * A feature of Java 8 that provides a way to process sequences of elements in a functional style. Supports operations like `filter`, `map`, `reduce`.

39. **What is the difference between intermediate and terminal operations in Streams?**

    * **Intermediate operations:** Return a new stream and are lazy (e.g., `filter`, `map`). **Terminal operations:** Produce a result or a side-effect and trigger the stream processing (e.g., `forEach`, `collect`).

40. **What is `Optional` class?**

    * A container object which may or may not contain a non-null value. Used to avoid `NullPointerException`.

41. **What is garbage collection?**

    * The process of automatically freeing memory by destroying objects that are no longer reachable.

42. **How can you request garbage collection?**

    * By calling `System.gc()`, but it's only a suggestion to the JVM and doesn't guarantee that it will run.

43. **What is reflection?**

* An API that allows inspecting and modifying the runtime behavior of applications. Can be used to examine classes, interfaces, fields, and methods at runtime.

44. **What are annotations?**

    * A form of metadata that can be added to Java source code. They can be processed at compile-time or runtime.

45. **What is serialization?**

    * The process of converting an object into a byte stream to store it or transmit it over a network.

46. **What is the `transient` keyword?**

    * Used to mark a field of an object to be excluded from serialization.

47. **What is the `Serializable` interface?**

    * A marker interface that a class must implement to be serializable.

48. **What is JDBC?**

    * Java Database Connectivity. An API for connecting and executing queries on a database.

49. **What are the steps to connect to a database in Java?**

    * 1. Load the driver. 2. Get a connection. 3. Create a statement. 4. Execute the query. 5. Process the result set. 6. Close the connection.

50. **What is a `PreparedStatement`?**

    * A pre-compiled SQL statement. It is more efficient and secure (prevents SQL injection) than a `Statement`.

51. **What is the difference between `equals()` and `hashCode()`?**

    * `hashCode()` returns an integer representation of the object's memory address. The contract is: if two objects are equal according to `equals()`, they must have the same hash code. The reverse is not necessarily true.

52. **What is the purpose of `hashCode()`?**

    * Used by hash-based collections like `HashMap` and `HashSet` to store and retrieve elements efficiently.

53. **What is a generic?**

    * A feature that allows you to define classes, interfaces, and methods with a placeholder for the type. Provides compile-time type safety.

54. **What is type erasure?**

    * The process by which the compiler removes generic type information at compile time and replaces it with casts.

55. **What is a wildcard in generics?**

    * Represented by `?`. Used to work with unknown types. Can be bounded (`? extends T` or `? super T`).

56. **What is an enum?**

    * A special data type that enables for a variable to be a set of predefined constants.

57. **What is autoboxing and unboxing?**

    * **Autoboxing:** The automatic conversion of a primitive type to its corresponding wrapper class object. **Unboxing:** The automatic conversion of a wrapper class object to its corresponding primitive type.

58. **What is the difference between `throw` and `throws`?**

    * `throw` is used to explicitly throw an exception. `throws` is used in a method signature to declare the exceptions that can be thrown by the method.

59. **What is the `ClassLoader`?**

    * A part of the JRE that dynamically loads Java classes into the JVM.

60. **What is the principle of 'write once, run anywhere'?**

    * Java code is compiled into platform-independent bytecode, which can be run on any machine that has a JVM.

61. **What is a JAR file?**

    * Java Archive. A package file format used to aggregate many files into one. It's a ZIP file with a `.jar` extension.

62. **What is the `main` method signature?**

    * `public static void main(String[] args)`.

Algorithms

1. **What is an algorithm?**
    - A step-by-step procedure or formula for solving a problem.
2. **What are the main characteristics of an algorithm?**
    - Input, Output, Finiteness, Definiteness, and Effectiveness.
3. **What is time complexity?**
    - The amount of time taken by an algorithm to run as a function of the length of the input.
4. **What is space complexity?**

- The amount of memory space required by an algorithm to run as a function of the length of the input.

5. **Explain Big O notation.**
    - A mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity. Used to classify algorithms according to their running time or space requirements.
6. **What is the difference between Big O, Big Omega, and Big Theta?**
    - **Big O:** Upper bound (worst-case). **Big Omega:** Lower bound (best-case). **Big Theta:** Tight bound (average-case).
7. **What is a sorting algorithm?**
    - An algorithm that puts elements of a list in a certain order.
8. **Explain Bubble Sort.**
    - A simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. Time complexity: O(n^2).
9. **Explain Selection Sort.**
    - A sorting algorithm that repeatedly finds the minimum element from the unsorted part and puts it at the beginning. Time complexity: O(n^2).
10. **Explain Insertion Sort.**

    * A sorting algorithm that builds the final sorted array one item at a time. It iterates through an input array and removes one element per iteration, finds the place the element belongs in the array, and then places it there. Time complexity: O(n^2).
11. **Explain Merge Sort.**

    * A divide and conquer algorithm that divides the array into two halves, recursively sorts them, and then merges the two sorted halves. Time complexity: O(n log n).
12. **Explain Quick Sort.**

    * A divide and conquer algorithm that picks an element as a pivot and partitions the given array around the picked pivot. Time complexity: O(n log n) on average, O(n^2) in the worst case.
13. **What is a stable sort?**

    * A sorting algorithm where two objects with equal keys appear in the same order in the sorted output as they appear in the input array.
14. **Which sorting algorithms are stable?**

    * Merge Sort, Insertion Sort, Bubble Sort.

15. **What is a searching algorithm?**

    * An algorithm used to find an element in a data structure.

16. **Explain Linear Search.**

    * A simple searching algorithm that sequentially checks each element of the list until a match is found or the whole list has been searched. Time complexity: O(n).

17. **Explain Binary Search.**

    * A searching algorithm that finds the position of a target value within a sorted array. It compares the target value to the middle element of the array. Time complexity: O(log n).

18. **What is a greedy algorithm?**

    * An algorithm that makes the locally optimal choice at each stage with the hope of finding a global optimum.

19. **Give an example of a greedy algorithm.**

    * Dijkstra's algorithm, Kruskal's algorithm, Prim's algorithm.

20. **What is a divide and conquer algorithm?**

    * An algorithm that recursively breaks down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.

21. **Give an example of a divide and conquer algorithm.**

    * Merge Sort, Quick Sort, Binary Search.

22. **What is dynamic programming?**

    * An optimization technique for solving complex problems by breaking them down into simpler subproblems and storing the results of subproblems to avoid re-computation.

23. **What are the two main properties of a problem that suggest dynamic programming?**

    * Optimal substructure and overlapping subproblems.

24. **Give an example of a dynamic programming problem.**

    * Fibonacci sequence, Knapsack problem, Longest Common Subsequence.

25. **What is the difference between divide and conquer and dynamic programming?**

    * Divide and conquer combines the solutions of subproblems to get the solution of the main problem, while dynamic programming uses the results of subproblems to find the optimal solution of the main problem. Dynamic programming is used when subproblems overlap.

26. **What is a backtracking algorithm?**

* A general algorithm for finding all (or some) solutions to some computational problems, notably constraint satisfaction problems, that incrementally builds candidates to the solutions, and abandons a candidate as soon as it determines that the candidate cannot possibly be completed to a valid solution.

27. **Give an example of a backtracking problem.**

    * N-Queens problem, Sudoku solver, maze solving.

28. **What is a randomized algorithm?**

    * An algorithm that employs a degree of randomness as part of its logic or procedure.

29. **What is a recursive algorithm?**

    * An algorithm that calls itself with smaller input values and which has a base case to terminate the recursion.

30. **What is the difference between iteration and recursion?**

    * Iteration uses loops to repeat a block of code. Recursion uses a function that calls itself. Recursion can be more elegant but may lead to stack overflow for large inputs.

31. **What is a hash function?**

    * A function that maps data of arbitrary size to a fixed-size value (hash). Used in hash tables.

32. **What are the properties of a good hash function?**

    * 1. Fast to compute. 2. Distributes keys uniformly. 3. Minimizes collisions.

33. **What is a collision in a hash table?**

    * When two different keys hash to the same index.

34. **How to handle collisions in a hash table?**

    * Chaining (using linked lists) and Open Addressing (probing for the next empty slot).

35. **What is a graph traversal algorithm?**

    * An algorithm for visiting all the nodes of a graph.

36. **Explain Breadth-First Search (BFS).**

    * A graph traversal algorithm that explores the neighbor nodes first, before moving to the next level neighbors. Uses a queue. Time complexity: O(V+E).

37. **Explain Depth-First Search (DFS).**

    * A graph traversal algorithm that explores as far as possible along each branch before backtracking. Uses a stack. Time complexity: O(V+E).

38. **What is Dijkstra's algorithm?**

* An algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It works for graphs with non-negative edge weights. Time complexity: O(E log V) with a priority queue.

39. **What is Bellman-Ford algorithm?**

    * An algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph. It is slower than Dijkstra's but can handle negative edge weights. Time complexity: O(V*E).

40. **What is Floyd-Warshall algorithm?**

    * An algorithm for finding shortest paths in a weighted graph with positive or negative edge weights (but with no negative cycles). It finds the shortest paths between all pairs of vertices. Time complexity: O(V^3).

41. **What is Kruskal's algorithm?**

    * A greedy algorithm for finding a minimum spanning tree (MST) of a connected, undirected graph. It finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. Time complexity: O(E log E).

42. **What is Prim's algorithm?**

    * A greedy algorithm for finding a minimum spanning tree (MST) of a connected, undirected graph. It starts from an arbitrary vertex and grows the MST by adding the cheapest edge connecting a vertex in the MST to a vertex outside the MST. Time complexity: O(E log V) with a priority queue.

43. **What is a minimum spanning tree (MST)?**

    * A subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

44. **What is the difference between Kruskal's and Prim's algorithm?**

    * Prim's algorithm grows the MST from a single vertex. Kruskal's algorithm selects edges from the entire graph.

45. **What is a topological sort?**

    * A linear ordering of the vertices of a directed acyclic graph (DAG) such that for every directed edge from vertex u to vertex v, u comes before v in the ordering. Time complexity: O(V+E).

46. **What is a string matching algorithm?**

    * An algorithm to find the occurrences of a pattern string within a larger text string.

47. **Explain the Naive string matching algorithm.**

* Slides the pattern one by one and checks for a match. Time complexity: O((n-m+1)*m).

48. **Explain the Knuth-Morris-Pratt (KMP) algorithm.**

    * An efficient string matching algorithm that preprocesses the pattern to create a partial match table (LPS array), which is used to skip characters when a mismatch occurs. Time complexity: O(n+m).

49. **What is the traveling salesman problem (TSP)?**

    * A classic NP-hard problem in which a salesman must find the shortest possible route that visits each city exactly once and returns to the origin city.

50. **What is the knapsack problem?**

    * A problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

51. **What is the difference between 0/1 knapsack and fractional knapsack?**

    * In 0/1 knapsack, items are indivisible; you either take an item or not. In fractional knapsack, you can take fractions of items.

52. **What is a randomized quicksort?**

    * A variation of quicksort where the pivot is chosen randomly, which helps to avoid the worst-case scenario.

53. **What is an approximation algorithm?**

    * An algorithm used to find an approximate solution to an optimization problem. Often used for NP-hard problems.

54. **What is a heuristic?**

    * A practical approach to problem solving that is not guaranteed to be optimal or perfect, but is sufficient for the immediate goals.

55. **What is the maximum flow problem?**

    * A problem to find the maximum amount of flow that can be sent from a source to a sink in a flow network.

56. **What is the Ford-Fulkerson algorithm?**

    * A greedy algorithm that computes the maximum flow in a flow network. It works by finding augmenting paths in the residual graph.

57. **What is a convex hull?**

    * The smallest convex polygon that contains all the given points in a plane.

58. **What is Graham scan?**

* An algorithm for finding the convex hull of a finite set of points in the plane. Time complexity: O(n log n).

59. **What is the closest pair of points problem?**

    * A problem of finding the two points in a set of n points that are closest to each other. A divide and conquer approach can solve it in O(n log n) time.

60. **What is a probabilistic algorithm?**

    * An algorithm that makes random choices at certain points. Examples include Monte Carlo and Las Vegas algorithms.

61. **What is the difference between a deterministic and a non-deterministic algorithm?**

    * A deterministic algorithm will always produce the same output for the same input. A non-deterministic algorithm may produce different outputs for the same input on different runs.

Data Structures

1. **What is a data structure?**
    - A way of organizing and storing data in a computer so that it can be accessed and modified efficiently.
2. **What is the difference between a linear and a non-linear data structure?**
    - **Linear:** Elements are arranged in a sequence (e.g., array, linked list). **Non-linear:** Elements are not arranged in a sequence (e.g., tree, graph).
3. **What is an array?**
    - A data structure that stores a fixed-size sequential collection of elements of the same type.
4. **What is a linked list?**
    - A linear data structure where elements are not stored at contiguous memory locations. Elements are linked using pointers.
5. **What are the advantages of a linked list over an array?**
    - Dynamic size, ease of insertion/deletion.
6. **What are the disadvantages of a linked list over an array?**
    - No random access (requires traversal), extra memory space for a pointer with each element.
7. **What are the types of linked lists?**
    - Singly linked list, Doubly linked list, Circular linked list.

8. **What is a doubly linked list?**
    - A linked list where each node has a pointer to the next node and a pointer to the previous node.
9. **What is a stack?**
    - A linear data structure that follows the Last-In-First-Out (LIFO) principle.
10. **What are the basic operations of a stack?**

    * `push` (add an element), `pop` (remove an element), `peek` (get the top element).
11. **What are the applications of a stack?**

    * Function calls (call stack), expression evaluation (infix to postfix), backtracking.
12. **What is a queue?**

    * A linear data structure that follows the First-In-First-Out (FIFO) principle.
13. **What are the basic operations of a queue?**

    * `enqueue` (add an element), `dequeue` (remove an element), `peek` (get the front element).
14. **What are the applications of a queue?**

    * CPU scheduling, handling requests on a single shared resource (e.g., printer), BFS.
15. **What is a priority queue?**

    * A special type of queue where each element has a priority. Elements with higher priority are served before elements with lower priority.
16. **What is a deque (double-ended queue)?**

    * A queue where elements can be added or removed from either end.
17. **What is a tree?**

    * A non-linear data structure that represents a hierarchical structure.
18. **What are the basic terminologies of a tree?**

    * Root, Node, Edge, Parent, Child, Leaf, Height, Depth.
19. **What is a binary tree?**

    * A tree data structure in which each node has at most two children, which are referred to as the left child and the right child.
20. **What is a binary search tree (BST)?**

    * A binary tree where the value of each node is greater than or equal to all values in its left subtree and less than all values in its right subtree.
21. **What are the advantages of a BST?**

* Fast search, insertion, and deletion (O(log n) on average).

22. **What is the worst-case time complexity for search in a BST?**

    * O(n) if the tree is skewed (unbalanced).

23. **What is a balanced binary search tree?**

    * A BST where the height of the two subtrees of every node never differ by more than one. Examples: AVL tree, Red-Black tree.

24. **What is an AVL tree?**

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

25. **What is a Red-Black tree?**

    * A self-balancing binary search tree where each node has an extra bit for storing color (red or black) and satisfies certain properties to ensure the tree remains balanced.

26. **What is a B-tree?**

    * A self-balancing tree data structure that maintains sorted data and allows searches, sequential access, insertions, and deletions in logarithmic time. It is optimized for systems that read and write large blocks of data, like databases and filesystems.

27. **What is a heap?**

    * A specialized tree-based data structure that satisfies the heap property. In a max heap, for any given node C, if P is a parent node of C, then the key of P is greater than or equal to the key of C. In a min heap, the key of P is less than or equal to the key of C.

28. **What are the applications of a heap?**

    * Heapsort, priority queue implementation, finding the k-th largest/smallest element.

29. **What is a graph?**

    * A non-linear data structure consisting of a set of vertices (nodes) and a set of edges that connect pairs of vertices.

30. **What is the difference between a directed and an undirected graph?**

    * In an undirected graph, edges have no orientation. In a directed graph, edges have a direction.

31. **What is a weighted graph?**

    * A graph where each edge has a numerical weight.

32. **How can you represent a graph?**

    * Adjacency matrix and Adjacency list.

33. **What is an adjacency matrix?**

* A 2D array where `adj[i][j] = 1` if there is an edge from vertex `i` to vertex `j`, otherwise `0`. For weighted graphs, it stores the weight.

34. **What is an adjacency list?**

    * An array of linked lists, where each index `i` in the array represents a vertex and the linked list at that index contains the vertices adjacent to `i`.

35. **Compare adjacency matrix and adjacency list.**

    * Adjacency matrix is faster for checking if an edge exists between two vertices (O(1)), but uses more space (O(V^2)). Adjacency list is more space-efficient for sparse graphs (O(V+E)) but slower for checking an edge (O(degree)).

36. **What is a hash table?**

    * A data structure that implements an associative array abstract data type, a structure that can map keys to values. It uses a hash function to compute an index into an array of buckets or slots, from which the desired value can be found.

37. **What is a trie (prefix tree)?**

    * A tree-like data structure that is used to store a dynamic set of strings. It is efficient for searching for words with a given prefix.

38. **What are the applications of a trie?**

    * Autocomplete, spell checkers, IP routing.

39. **What is a segment tree?**

    * A tree data structure for storing information about intervals, or segments. It allows querying which of the stored segments contain a given point.

40. **What is a Fenwick tree (Binary Indexed Tree)?**

    * A data structure that can efficiently update elements and calculate prefix sums in a table of numbers.

41. **What is a disjoint set union (DSU) or Union-Find?**

    * A data structure that keeps track of a set of elements partitioned into a number of disjoint (non-overlapping) subsets. It has two primary operations: `find` (determine which subset an element is in) and `union` (join two subsets into a single subset).

42. **What are the optimizations for DSU?**

    * Path compression and Union by rank/size.

43. **What is a sparse table?**

    * A data structure that allows answering range queries in O(1) time after an O(n log n) preprocessing.

44. **What is a suffix tree?**

* A compressed trie containing all the suffixes of the given text as their keys and positions in the text as their values. Used for fast string searching.

45. **What is a suffix array?**

    * A sorted array of all suffixes of a string. It is a more space-efficient alternative to a suffix tree.

46. **What is a Cartesian tree?**

    * A binary tree derived from a sequence of numbers. It has the heap property and the inorder traversal of the tree returns the original sequence.

47. **What is a k-d tree?**

    * A space-partitioning data structure for organizing points in a k-dimensional space. Used for range searches and nearest neighbor searches.

48. **What is a Bloom filter?**

    * A space-efficient probabilistic data structure that is used to test whether an element is a member of a set. False positive matches are possible, but false negatives are not.

49. **What is a skip list?**

    * A probabilistic data structure that allows for fast search within an ordered sequence of elements. It is built in layers, with each layer being a subset of the previous layer.

50. **What is a Fibonacci heap?**

    * A data structure for priority queue operations, consisting of a collection of heap-ordered trees. It has a better amortized running time than a binary heap.

51. **What is a scapegoat tree?**

    * A self-balancing binary search tree that is easy to implement and does not require extra storage for balance information. It rebalances by completely rebuilding a subtree into a perfectly balanced tree when an insertion makes the tree too unbalanced.

52. **What is a treap?**

    * A randomized binary search tree that maintains dynamic set of ordered keys and allows binary searches among the keys. Each node in the tree has a key and a priority, where the keys are in sorted order and the priorities satisfy the heap property.

53. **What is a rope?**

    * A tree data structure composed of smaller strings that is used for storing and manipulating a very long string. It allows for faster concatenation and splitting of strings than a contiguous array.

54. **What is a circular buffer (ring buffer)?**

    * A fixed-size buffer as if it were connected end-to-end. Useful for buffering data streams.

55. **What is a gap buffer?**

    * A dynamic array that allows for efficient insertion and deletion operations clustered together. It is commonly used in text editors.

56. **What is a Judy array?**

    * A highly efficient and memory-sparing data structure for storing and looking up string keys.

57. **What is a van Emde Boas tree?**

    * A tree data structure which implements an associative array with m-bit integer keys. It performs all operations in O(log log M) time, where M is the maximum number of elements that can be stored.

58. **What is a Radix tree (Patricia trie)?**

    * A space-optimized trie in which nodes with only one child are merged with their parent.

59. **What is a ternary search tree?**

    * A type of trie where nodes have three children (less than, equal to, and greater than). It is more space-efficient than a standard trie for storing strings.

60. **What is a quadtree?**

    * A tree data structure in which each internal node has exactly four children. Quadtrees are most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions.

61. **What is an octree?**

    * A tree data structure in which each internal node has exactly eight children. Octrees are most often used to partition a three-dimensional space by recursively subdividing it into eight octants.

Data Mining

1. **What is Data Mining?**
    - The process of discovering patterns and insights from large datasets.
2. **What are the main tasks of Data Mining?**
    - Classification, Regression, Clustering, Association Rule Mining, Anomaly Detection.
3. **What is supervised learning?**
    - A type of machine learning where the model is trained on labeled data (input-output pairs).

4. **What is unsupervised learning?**
    - A type of machine learning where the model is trained on unlabeled data, finding patterns without explicit guidance.
5. **What is classification?**
    - A data mining task that assigns items in a collection to target categories or classes.
6. **Give examples of classification algorithms.**
    - Decision Trees, Support Vector Machines (SVM), Naive Bayes, K-Nearest Neighbors (KNN).
7. **What is regression?**
    - A data mining task that predicts a continuous target variable based on input features.
8. **Give examples of regression algorithms.**
    - Linear Regression, Polynomial Regression, Decision Tree Regression.
9. **What is clustering?**
    - A data mining task that groups a set of objects in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups.
10. **Give examples of clustering algorithms.**

    * K-Means, Hierarchical Clustering, DBSCAN.
11. **What is association rule mining?**

    * A data mining technique that discovers interesting relationships or associations among a set of items in a dataset.
12. **Explain the Apriori algorithm.**

    * An algorithm for frequent itemset mining and association rule learning over transactional databases. It identifies frequent individual items in the database and extends them to larger itemsets.
13. **What is anomaly detection?**

    * The identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.
14. **What is data preprocessing?**

    * The process of transforming raw data into an understandable format for data mining. Includes cleaning, integration, transformation, reduction.
15. **Why is data cleaning important?**

    * To handle missing values, noisy data, and inconsistencies, improving data quality.

16. **What are ways to handle missing values?**

    * Imputation (mean, median, mode), deletion of rows/columns, using algorithms that handle missing values.

17. **What is data transformation?**

    * The process of converting data from one format or structure into another. Includes normalization, aggregation, generalization.

18. **Explain normalization in data mining.**

    * Scaling numerical data to a standard range (e.g., 0-1) to prevent features with larger values from dominating.

19. **What is data reduction?**

    * Reducing the volume of data while producing the same or similar analytical results. Includes dimensionality reduction, numerosity reduction.

20. **What is dimensionality reduction?**

    * Reducing the number of random variables under consideration. Examples: PCA, feature selection.

21. **Explain PCA (Principal Component Analysis).**

    * A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components.

22. **What is feature selection?**

    * The process of selecting a subset of relevant features (variables) for use in model construction.

23. **What is a decision tree?**

    * A flowchart-like structure where each internal node represents a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a class label.

24. **What is overfitting in data mining?**

    * When a model learns the training data too well, including noise and outliers, leading to poor performance on unseen data.

25. **How to prevent overfitting?**

    * Cross-validation, regularization, pruning (for decision trees), increasing training data, early stopping.

26. **What is cross-validation?**

* A technique to evaluate the performance of a model by partitioning the data into subsets, training on some, and testing on others.

27. **Explain K-fold cross-validation.**

    * The dataset is divided into K equal folds. The model is trained on K-1 folds and tested on the remaining fold. This process is repeated K times.

28. **What is a confusion matrix?**

    * A table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

29. **Define Accuracy, Precision, Recall, F1-Score.**

    * **Accuracy:** (TP+TN)/(TP+TN+FP+FN). **Precision:** TP/(TP+FP). **Recall:** TP/(TP+FN). **F1-Score:** Harmonic mean of Precision and Recall.

30. **What is a ROC curve?**

    * Receiver Operating Characteristic curve. A graph showing the performance of a classification model at all classification thresholds.

31. **What is AUC?**

    * Area Under the ROC Curve. Measures the entire 2-D area underneath the entire ROC curve. Represents the degree or measure of separability.

32. **What is a Support Vector Machine (SVM)?**

    * A supervised learning model that finds an optimal hyperplane that best separates data points into different classes.

33. **What is a kernel trick in SVM?**

    * A method that allows SVMs to operate in a high-dimensional, implicit feature space without ever explicitly computing the coordinates of the data in that space.

34. **What is Naive Bayes?**

    * A probabilistic classifier based on Bayes theorem with the 'naive' assumption of independence between features.

35. **What is K-Nearest Neighbors (KNN)?**

    * A non-parametric, lazy learning algorithm that classifies a data point based on how its neighbors are classified.

36. **What is DBSCAN?**

    * Density-Based Spatial Clustering of Applications with Noise. A clustering algorithm that groups together points that are closely packed together, marking as outliers points that lie alone in low-density regions.

37. **What is hierarchical clustering?**

* A clustering algorithm that builds a hierarchy of clusters. Can be agglomerative (bottom-up) or divisive (top-down).

38. **What is a frequent itemset?**

    * A set of items that appears frequently in a transaction dataset.

39. **What is support, confidence, and lift in association rules?**

    * **Support:** Frequency of an itemset. **Confidence:** Conditional probability of consequent given antecedent. **Lift:** Measures how many times more often X and Y occur together than expected if they were statistically independent.

40. **What is market basket analysis?**

    * A data mining technique used to discover associations between items that are frequently purchased together.

41. **What is collaborative filtering?**

    * A technique used by recommender systems that makes predictions about the interests of a user by collecting preferences from many users.

42. **What is content-based filtering?**

    * A recommender system that recommends items similar to those a user has liked in the past.

43. **What is data warehousing?**

    * A system used for reporting and data analysis, and is considered a core component of business intelligence.

44. **What is OLAP (Online Analytical Processing)?**

    * A computer-based technique of analyzing business data for decision-making purposes. Allows for multi-dimensional analysis.

45. **What is OLTP (Online Transaction Processing)?**

    * A type of data processing that facilitates and manages transaction-oriented applications, typically for data entry and retrieval transaction processing.

46. **What is ETL?**

    * Extract, Transform, Load. A process in data warehousing that extracts data from source systems, transforms it into a consistent format, and loads it into a data warehouse.

47. **What is a data cube?**

    * A multi-dimensional array of data, used for OLAP operations.

48. **What is data generalization?**

    * Summarizing data by replacing low-level data with higher-level concepts.

49. **What is data discretization?**

    * Dividing the range of a continuous attribute into intervals.

50. **What is a data stream?**

    * A continuous flow of data that needs to be processed incrementally.

51. **What is concept drift?**

    * When the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways.

52. **What is active learning?**

    * A machine learning approach where the learning algorithm can interactively query a user or some other information source to label new data points.

53. **What is ensemble learning?**

    * A technique that combines multiple machine learning models to obtain better predictive performance than could be obtained from any of the constituent models alone.

54. **Explain Bagging (Bootstrap Aggregating).**

    * An ensemble method that trains multiple models on different bootstrap samples of the training data and averages their predictions.

55. **Explain Boosting.**

    * An ensemble method that sequentially builds models, where each new model attempts to correct the errors of the previous ones.

56. **Give examples of Boosting algorithms.**

    * AdaBoost, Gradient Boosting (GBM), XGBoost, LightGBM.

57. **What is Random Forest?**

    * An ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

58. **What is the curse of dimensionality?**

    * Various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings.

59. **What is a recommender system?**

    * A subclass of information filtering system that seeks to predict the 'rating' or 'preference' a user would give to an item.

60. **What is a time series?**

* A sequence of data points indexed (or listed or graphed) in time order.

61. **What is trend, seasonality, and cycle in time series?**

    * **Trend:** Long-term increase or decrease. **Seasonality:** Regular, predictable patterns that recur over a fixed period. **Cycle:** Patterns that are not of fixed duration.

62. **What is a data lake?**

    * A centralized repository that allows you to store all your structured and unstructured data at any scale.

63. **What is a data warehouse?**

    * A large collection of business data used to help an organization make decisions. It is a central repository of integrated data from one or more disparate sources.

64. **Difference between data lake and data warehouse?**

    * Data lakes store raw, unstructured data for future use, while data warehouses store structured, processed data for immediate analysis.

65. **What is CRISP-DM?**

    * Cross-Industry Standard Process for Data Mining. A data mining methodology that describes common approaches used by data mining experts.

66. **What are the phases of CRISP-DM?**

    * Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment.

Operating Systems

1. **What is an Operating System?**
    - A software that manages computer hardware and software resources and provides common services for computer programs.
2. **What are the main functions of an Operating System?**
    - Process management, memory management, file management, I/O management, security, networking.
3. **What is a process?**
    - A program in execution. It is an active entity, unlike a program which is a passive entity.
4. **What is a thread?**
    - A lightweight process. It is a basic unit of CPU utilization, consisting of a program counter, a stack, and a set of registers.

5. **Difference between process and thread?**
    - Processes are independent, have their own memory space, and inter-process communication is complex. Threads share the same memory space, are lighter, and inter-thread communication is simpler.
6. **What is a PCB (Process Control Block)?**
    - A data structure containing information about a process, such as process state, program counter, CPU registers, memory management information, etc.
7. **What are the different states of a process?**
    - New, Ready, Running, Waiting, Terminated.
8. **What is context switching?**
    - The process of saving the state of one process and restoring the state of another process so that execution can be resumed from the same point later.
9. **What is a scheduler?**
    - A component of the OS that selects which process to run next.
10. **What are the types of schedulers?**

    * Long-term (job scheduler), Short-term (CPU scheduler), Medium-term (swapping).
11. **Explain different CPU scheduling algorithms.**

    * **FCFS (First-Come, First-Served):** Non-preemptive, simple. **SJF (Shortest Job First):** Optimal for minimum average waiting time, can be preemptive or non-preemptive. **Priority Scheduling:** Processes with higher priority run first. **Round Robin:** Preemptive, each process gets a small time quantum.
12. **What is a deadlock?**

    * A situation where two or more processes are blocked indefinitely, waiting for each other to release resources.
13. **What are the four necessary conditions for a deadlock?**

    * Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait.
14. **How to handle deadlocks?**

    * Prevention, Avoidance, Detection and Recovery.
15. **Explain deadlock prevention.**

    * Ensuring that at least one of the four necessary conditions for deadlock cannot hold.
16. **Explain deadlock avoidance (Banker's Algorithm).**

    * Dynamically checking the resource-allocation state to ensure that there can never be a circular-wait condition.

17. **What is starvation?**

    * A situation where a process is repeatedly denied access to a resource or CPU, even though the resource becomes available.

18. **What is a semaphore?**

    * A synchronization tool used to control access to a common resource in a concurrent system. It is an integer variable that is accessed only through two standard atomic operations: `wait()` (or `P()`) and `signal()` (or `V()`).

19. **What is a mutex?**

    * A binary semaphore that provides mutual exclusion. Only one thread can acquire the mutex at a time.

20. **Difference between semaphore and mutex?**

    * Semaphore is a signaling mechanism, mutex is a locking mechanism. Semaphore can be used for more than one resource, mutex for a single resource.

21. **What is a monitor?**

    * A high-level synchronization construct that allows processes to safely access shared resources. It consists of a mutex and condition variables.

22. **What is critical section?**

    * A segment of code that accesses shared resources and must be executed by only one process at a time.

23. **What is race condition?**

    * A situation where multiple processes or threads try to access and modify the same shared data concurrently, and the outcome depends on the relative order of execution.

24. **What is memory management?**

    * The process of controlling and coordinating computer memory, assigning memory blocks to running programs, and optimizing overall system performance.

25. **What is paging?**

    * A memory management scheme that allows the physical address space of a process to be non-contiguous. It divides physical memory into fixed-size blocks called frames and logical memory into blocks of the same size called pages.

26. **What is segmentation?**

    * A memory management scheme that supports user view of memory. Memory is divided into segments of varying sizes, each corresponding to a logical unit of the program.

27. **Difference between paging and segmentation?**

* Paging divides memory into fixed-size pages, segmentation into variable-size segments. Paging is invisible to the user, segmentation is visible.

28. **What is virtual memory?**

    * A memory management technique that allows a computer to compensate for physical memory shortages by temporarily transferring data from RAM to disk storage.

29. **What is demand paging?**

    * A virtual memory technique where pages are loaded into memory only when they are needed.

30. **What is a page fault?**

    * An event that occurs when a program tries to access a page that is currently not in physical memory.

31. **Explain different page replacement algorithms.**

    * **FIFO (First-In, First-Out):** Replaces the oldest page. **LRU (Least Recently Used):** Replaces the page that has not been used for the longest period of time. **Optimal:** Replaces the page that will not be used for the longest period of time (ideal, not practical).

32. **What is thrashing?**

    * A condition where the system spends most of its time swapping pages in and out of memory, rather than executing application code.

33. **What is external fragmentation?**

    * Total memory space exists to satisfy a request, but it is not contiguous.

34. **What is internal fragmentation?**

    * Memory that is internal to a partition but is not being used.

35. **What is a file system?**

    * A method and data structure that an operating system uses to control how data is stored and retrieved.

36. **What are the different file access methods?**

    * Sequential access, Direct access, Indexed sequential access.

37. **What is disk scheduling?**

    * The process of deciding the order in which disk I/O requests are serviced.

38. **Explain different disk scheduling algorithms.**

    * **FCFS (First-Come, First-Served):** Simplest. **SSTF (Shortest Seek Time First):** Prioritizes requests closest to the current head position. **SCAN (Elevator Algorithm):** Head moves in one direction, servicing requests, then reverses. **C-SCAN (Circular**

SCAN): Head moves in one direction, servicing requests, then returns to the other end without servicing requests on the return trip.

39. **What is RAID?**

    * Redundant Array of Independent Disks. A data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for the purposes of data redundancy, performance improvement, or both.

40. **What is a kernel?**

    * The core component of an operating system that manages system resources and provides services to applications.

41. **Difference between monolithic and microkernel?**

    * **Monolithic kernel:** All OS services run in kernel space. **Microkernel:** Only essential services run in kernel space, others in user space.

42. **What is a system call?**

    * A programmatic way in which a computer program requests a service from the kernel of the operating system.

43. **What is an interrupt?**

    * A signal to the processor emitted by hardware or software indicating an event that needs immediate attention.

44. **What is polling?**

    * The process of actively sampling the status of an external device by a client program as a synchronous activity.

45. **Difference between interrupt and polling?**

    * Interrupts are asynchronous and event-driven, polling is synchronous and continuously checks status.

46. **What is buffering?**

    * A technique of storing data temporarily in a buffer while it is being transferred from one place to another.

47. **What is spooling?**

    * Simultaneous Peripheral Operations Online. A process in which data is temporarily held to be used and executed by a device, program or the system.

48. **What is a shell?**

    * A user interface for access to an operating system's services. It can be a command-line interface (CLI) or a graphical user interface (GUI).

49. **What is a zombie process?**

* A process that has completed execution but still has an entry in the process table because its parent process has not yet read its exit status.

50. **What is an orphan process?**

    * A process whose parent process has terminated before it has. It is then adopted by the `init` process.

51. **What is a daemon?**

    * A computer program that runs as a background process, rather than being under the direct control of an interactive user.

52. **What is client-server model?**

    * A distributed application architecture that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

53. **What is peer-to-peer model?**

    * A decentralized communications model in which each party has equivalent capabilities and can initiate a communication session.

54. **What is distributed operating system?**

    * An operating system that manages a group of independent computers and makes them appear to be a single computer.

55. **What is real-time operating system (RTOS)?**

    * An operating system intended for applications with strict time constraints. It processes data as it comes in, typically without buffer delays.

56. **What is a hypervisor?**

    * A software, firmware, or hardware that creates and runs virtual machines (VMs). Also known as a Virtual Machine Monitor (VMM).

57. **What is virtualization?**

    * The creation of a virtual (rather than actual) version of something, such as a server, a storage device, a network resource, or even an operating system.

58. **What is containerization?**

    * A form of operating system virtualization in which an application or service is packaged into a container image, which can then be deployed and run in an isolated environment.

59. **Difference between virtualization and containerization?**

    * Virtualization virtualizes the hardware, running multiple OS instances. Containerization virtualizes the OS, sharing the host OS kernel.

60. **What is a bootloader?**

* A program that loads an operating system or runtime environment when a computer is started.

61. **What is BIOS/UEFI?**

    * **BIOS (Basic Input/Output System):** Firmware used to perform hardware initialization during the booting process. **UEFI (Unified Extensible Firmware Interface):** A modern replacement for BIOS.

62. **What is an inode?**

    * A data structure in a Unix-style file system that describes a file-system object such as a file or a directory. Each inode stores the attributes and disk block locations of the object's data.

63. **What is a superblock?**

    * A data structure that contains information about the file system as a whole, such as its size, the number of free blocks, and the location of the inode table.

64. **What is journaling file system?**

    * A file system that keeps track of changes not yet committed to the main file system by recording them in a data structure called a journal.

65. **What is a memory-mapped file?**

    * A segment of virtual memory that has been assigned a direct byte-for-byte correlation with some portion of a file or file-like resource.

66. **What is copy-on-write (COW)?**

    * A resource-management technique used in computer programming to efficiently duplicate modifiable resources. If a resource is duplicated but not modified, both copies share the same underlying resource.

67. **What is the difference between user mode and kernel mode?**

    * **User mode:** Restricted access to system resources. **Kernel mode:** Full access to system hardware and memory.

68. **What is a system call table?**

    * A table that maps system call numbers to the corresponding kernel functions.

69. **What is the role of the MMU (Memory Management Unit)?**

    * A computer hardware component that handles memory access requests originated by the CPU. It translates virtual addresses to physical addresses.

70. **What is the difference between logical and physical address?**

    * **Logical address:** Address generated by the CPU. **Physical address:** Actual address in main memory.

71. **What is a page table?**

    * A data structure used by a virtual memory system in a computer operating system to store the mapping between virtual addresses and physical addresses.

72. **What is a TLB (Translation Lookaside Buffer)?**

    * A CPU cache that stores recent translations of virtual memory to physical addresses, used to speed up virtual address translation.

73. **What is a device driver?**

    * A computer program that operates or controls a particular type of device that is attached to a computer.

74. **What is a boot sector?**

    * A sector on a data storage device that contains machine code to be loaded into random-access memory (RAM) by a computer system's built-in firmware (e.g., BIOS) and executed to start the boot process.

75. **What is a process synchronization?**

    * The task of coordinating the execution of multiple processes in a way that no two processes can access the same shared resource and data.

76. **What is the Dining Philosophers Problem?**

    * A classic synchronization problem illustrating the challenges of avoiding deadlock and starvation when multiple processes compete for limited resources.

77. **What is the Producer-Consumer Problem?**

    * A classic synchronization problem where a producer process generates data and a consumer process consumes it, with a shared buffer between them.

78. **What is the Readers-Writers Problem?**

    * A classic synchronization problem where multiple readers can access a shared resource concurrently, but only one writer can access it at a time.

Software Design

1. **What is Software Design?**
    - The process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.
2. **What are the key principles of good software design?**
    - Modularity, Abstraction, Encapsulation, Low Coupling, High Cohesion, Reusability, Flexibility, Scalability, Maintainability.

3. **Explain Modularity.**
    - Breaking down a software system into smaller, independent, and interchangeable components (modules).
4. **Explain Abstraction.**
    - Hiding complex implementation details and showing only essential features to the user.
5. **Explain Encapsulation.**
    - Bundling data and methods that operate on the data within a single unit (class), restricting direct access to some of the object's components.
6. **What is Coupling?**
    - The degree of interdependence between software modules. Low coupling is desirable.
7. **What is Cohesion?**
    - The degree to which the elements within a module belong together. High cohesion is desirable.
8. **What are Design Patterns?**
    - Reusable solutions to common problems in software design. They are not direct solutions but templates that can be adapted.
9. **Give examples of Creational Design Patterns.**
    - Singleton, Factory Method, Abstract Factory, Builder, Prototype.
10. **Give examples of Structural Design Patterns.**

    * Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy.
11. **Give examples of Behavioral Design Patterns.**

    * Chain of Responsibility, Command, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor.
12. **Explain the Singleton Design Pattern.**

    * Ensures a class has only one instance and provides a global point of access to that instance.
13. **Explain the Factory Method Design Pattern.**

    * Defines an interface for creating an object, but lets subclasses decide which class to instantiate.
14. **Explain the Observer Design Pattern.**

* Defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

15. **Explain the Strategy Design Pattern.**

    * Defines a family of algorithms, encapsulates each one, and makes them interchangeable. Strategy lets the algorithm vary independently from clients that use it.

16. **What is SOLID principles?**

    * A set of five design principles intended to make software designs more understandable, flexible, and maintainable.

17. **Explain S in SOLID (Single Responsibility Principle).**

    * A class should have only one reason to change, meaning it should have only one job or responsibility.

18. **Explain O in SOLID (Open/Closed Principle).**

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

19. **Explain L in SOLID (Liskov Substitution Principle).**

    * Objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.

20. **Explain I in SOLID (Interface Segregation Principle).**

    * Clients should not be forced to depend on interfaces they do not use. Better to have many small, specific interfaces than one large, general-purpose interface.

21. **Explain D in SOLID (Dependency Inversion Principle).**

    * High-level modules should not depend on low-level modules. Both should depend on abstractions. Abstractions should not depend on details. Details should depend on abstractions.

22. **What is MVC (Model-View-Controller)?**

    * A software architectural pattern for implementing user interfaces, dividing an application into three interconnected components.

23. **Explain Model in MVC.**

    * Represents the application's data, business logic, and rules.

24. **Explain View in MVC.**

    * Presents the data to the user. It is responsible for the look and feel.

25. **Explain Controller in MVC.**

    * Handles user input, manipulates the model, and updates the view.

26. **What is an API (Application Programming Interface)?**

    * A set of definitions and protocols for building and integrating application software. It specifies how software components should interact.

27. **What is REST (Representational State Transfer)?**

    * An architectural style for designing networked applications. It relies on a stateless, client-server, cacheable communications protocol (typically HTTP).

28. **What are the characteristics of RESTful APIs?**

    * Stateless, Client-Server, Cacheable, Layered System, Uniform Interface.

29. **What is SOAP (Simple Object Access Protocol)?**

    * A messaging protocol specification for exchanging structured information in the implementation of web services.

30. **Difference between REST and SOAP?**

    * REST is an architectural style, SOAP is a protocol. REST is typically simpler and uses HTTP, SOAP is more rigid and uses XML.

31. **What is Microservices Architecture?**

    * An architectural style that structures an application as a collection of loosely coupled services, each running in its own process and communicating with lightweight mechanisms.

32. **What are the advantages of Microservices?**

    * Scalability, resilience, independent deployment, technology diversity.

33. **What are the disadvantages of Microservices?**

    * Complexity, distributed data management, inter-service communication overhead.

34. **What is Monolithic Architecture?**

    * A traditional architectural style where all components of an application are tightly coupled and run as a single service.

35. **Difference between Monolithic and Microservices?**

    * Monolithic is a single, tightly coupled unit. Microservices are small, independent, loosely coupled services.

36. **What is Domain-Driven Design (DDD)?**

    * An approach to software development that centers on programming a complex system by focusing on the core domain logic.

37. **What is a Bounded Context in DDD?**

* A central pattern in DDD. It defines the boundaries within which a particular domain model is applicable.

38. **What is Event-Driven Architecture (EDA)?**

    * A software architecture pattern promoting the production, detection, consumption of, and reaction to events.

39. **What is a Message Queue?**

    * A form of asynchronous service-to-service communication used in serverless and microservices architectures. Messages are stored until they are processed and deleted.

40. **What is a Load Balancer?**

    * A device that distributes network traffic evenly across multiple servers to ensure high availability and reliability.

41. **What is a Proxy Server?**

    * A server that acts as an intermediary for requests from clients seeking resources from other servers.

42. **What is a Reverse Proxy?**

    * A type of proxy server that retrieves resources on behalf of a client from one or more servers.

43. **What is Caching?**

    * Storing copies of data in a temporary storage location (cache) so that future requests for that data can be served faster.

44. **What is CDN (Content Delivery Network)?**

    * A geographically distributed network of proxy servers and their data centers. The goal is to provide high availability and performance by distributing the service spatially relative to end-users.

45. **What is Database Sharding?**

    * A method of distributing a single logical dataset into multiple databases. Each shard (database partition) is independent and self-contained.

46. **What is Database Replication?**

    * The process of creating and maintaining multiple copies of a database to improve availability, fault tolerance, and performance.

47. **What is eventual consistency?**

    * A consistency model used in distributed systems where, if no new updates are made to a given data item, all reads of that item will eventually return the last updated value.

48. **What is CAP Theorem?**

* A theorem stating that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees: Consistency, Availability, and Partition tolerance.

49. **What is Idempotence?**

    * The property of certain operations in mathematics and computer science whereby they can be applied multiple times without changing the result beyond the initial application.

50. **What is a Circuit Breaker pattern?**

    * A design pattern used in distributed systems to prevent a cascading failure when a service is unavailable or performing poorly.

51. **What is a Saga pattern?**

    * A way to manage distributed transactions in microservices architecture. A saga is a sequence of local transactions where each transaction updates data within a single service.

52. **What is Feature Toggling (Feature Flags)?**

    * A technique in software development that allows teams to turn features on or off during runtime without deploying new code.

53. **What is A/B Testing?**

    * A method of comparing two versions of a webpage or app against each other to determine which one performs better.

54. **What is Continuous Integration (CI)?**

    * The practice of merging all developers' working copies to a shared mainline several times a day.

55. **What is Continuous Delivery (CD)?**

    * An extension of CI to ensure that software can be released to production at any time.

56. **What is DevOps?**

    * A set of practices that combines software development (Dev) and IT operations (Ops) to shorten the systems development life cycle and provide continuous delivery with high software quality.

57. **What is Infrastructure as Code (IaC)?**

    * Managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

58. **What is a Container Orchestration?**

* The automated arrangement, coordination, and management of computer containers. Examples: Kubernetes, Docker Swarm.

59. **What is Serverless Architecture?**

    * A cloud-native development model that allows developers to build and run applications without having to manage servers.

60. **What is a Message Broker?**

    * An intermediary computer program module that translates a message from the formal messaging protocol of the sender to the formal messaging protocol of the receiver.

61. **What is an API Gateway?**

    * A server that acts as an API front-end, takes a single entry point for all clients, and routes requests to the appropriate microservice or backend service.

62. **What is a Service Mesh?**

    * A dedicated infrastructure layer for handling service-to-service communication. It ensures that communication among services is fast, reliable, and secure.

63. **What is Observability in software?**

    * The ability to measure the internal states of a system by examining its outputs. It is achieved through logging, metrics, and tracing.

64. **What is Telemetry?**

    * The in situ collection of measurements or other data at remote points and their automatic transmission to receiving equipment for monitoring.

65. **What is a Data Pipeline?**

    * A series of data processing steps that move data from source systems to target systems.

66. **What is a Data Lakehouse?**

    * A new open architecture that combines the best elements of data lakes and data warehouses.

67. **What is a Data Mesh?**

    * A decentralized data architecture that organizes data by specific business domains, treating data as a product.

Pattern Recognition

1. **What is Pattern Recognition?**

    - The automated discovery of patterns and regularities in data using computer algorithms and the appropriate choice of data representation.

2. **What is the difference between Pattern Recognition and Machine Learning?**
    - Pattern Recognition is often considered a subfield of Machine Learning, focusing on the classification of data based on patterns. ML is broader, including prediction, regression, etc.
3. **What are the main components of a pattern recognition system?**
    - Sensor, Preprocessing, Feature Extraction, Feature Selection, Classifier/Recognizer, Post-processing.
4. **What is feature extraction?**
    - The process of transforming raw data into a set of features that are informative and non-redundant, facilitating the subsequent learning and generalization steps.
5. **What is feature selection?**
    - The process of selecting a subset of relevant features for use in model construction, reducing dimensionality and improving model performance.
6. **What is a classifier?**
    - An algorithm that maps input data to a category or class label.
7. **What is supervised pattern recognition?**
    - When the training data is labeled with known classes, and the system learns a mapping from features to classes.
8. **What is unsupervised pattern recognition?**
    - When the training data is unlabeled, and the system tries to find inherent structures or groupings in the data (e.g., clustering).
9. **What is semi-supervised learning?**
    - A type of machine learning that uses a small amount of labeled data with a large amount of unlabeled data during training.
10. **What is a decision boundary?**

    * A hypersurface that partitions the underlying vector space into two or more regions, one for each class.
11. **What is a linear classifier?**

    * A classifier that makes its classification decision based on a linear combination of its input features.
12. **Give examples of linear classifiers.**

    * Perceptron, Linear SVM, Logistic Regression.
13. **What is a non-linear classifier?**

* A classifier that uses a non-linear decision boundary to separate classes.

14. **Give examples of non-linear classifiers.**

    * Kernel SVM, Decision Trees, Neural Networks.

15. **What is a confusion matrix in pattern recognition?**

    * A table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

16. **What is a false positive and false negative?**

    * **False Positive (Type I error):** Model predicts positive, but it's actually negative. **False Negative (Type II error):** Model predicts negative, but it's actually positive.

17. **What is precision and recall?**

    * **Precision:** Proportion of true positives among all positive predictions. **Recall:** Proportion of true positives among all actual positives.

18. **What is the F1-score?**

    * The harmonic mean of precision and recall, providing a single metric that balances both.

19. **What is a ROC curve and AUC?**

    * **ROC (Receiver Operating Characteristic) curve:** Plots the True Positive Rate (Recall) against the False Positive Rate at various threshold settings. **AUC (Area Under the Curve):** Measures the entire 2-D area underneath the entire ROC curve, indicating the model's ability to distinguish between classes.

20. **What is a perceptron?**

    * The simplest form of an artificial neural network, a linear classifier for binary classification.

21. **What is the perceptron learning algorithm?**

    * An algorithm for training a binary classifier called a perceptron. It iteratively adjusts weights based on misclassified samples.

22. **What is the limitation of a single-layer perceptron?**

    * It can only classify linearly separable data.

23. **What is a multi-layer perceptron (MLP)?**

    * A feedforward artificial neural network that consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. It can classify non-linearly separable data.

24. **What is backpropagation?**

* A supervised learning algorithm for training artificial neural networks. It calculates the gradient of the loss function with respect to the weights of the network.

25. **What is a support vector machine (SVM)?**

    * A supervised learning model that finds an optimal hyperplane that best separates data points into different classes, maximizing the margin between the classes.

26. **What is the kernel trick in SVM?**

    * A method that allows SVMs to operate in a high-dimensional, implicit feature space without ever explicitly computing the coordinates of the data in that space.

27. **What is K-Nearest Neighbors (KNN) in pattern recognition?**

    * A non-parametric, lazy learning algorithm that classifies a data point based on the majority class of its K nearest neighbors in the feature space.

28. **What is a decision tree in pattern recognition?**

    * A flowchart-like structure where each internal node represents a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a class label.

29. **What is ensemble learning?**

    * A technique that combines multiple machine learning models to obtain better predictive performance than could be obtained from any of the constituent models alone.

30. **What is bagging and boosting?**

    * **Bagging:** Trains multiple models independently on different subsets of data and averages their predictions. **Boosting:** Sequentially builds models, where each new model tries to correct errors of previous ones.

31. **What is Random Forest?**

    * An ensemble learning method that constructs a multitude of decision trees at training time and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees.

32. **What is clustering in pattern recognition?**

    * The task of grouping a set of objects in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups.

33. **What is K-Means clustering?**

    * An unsupervised learning algorithm that partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

34. **What is hierarchical clustering?**

* A clustering algorithm that builds a hierarchy of clusters. Can be agglomerative (bottom-up) or divisive (top-down).

35. **What is DBSCAN?**

    * Density-Based Spatial Clustering of Applications with Noise. A clustering algorithm that groups together points that are closely packed together, marking as outliers points that lie alone in low-density regions.

36. **What is principal component analysis (PCA)?**

    * A statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components.

37. **What is Linear Discriminant Analysis (LDA)?**

    * A dimensionality reduction technique used to find a linear combination of features that characterizes or separates two or more classes of objects or events.

38. **Difference between PCA and LDA?**

    * PCA is an unsupervised method that finds directions of maximum variance. LDA is a supervised method that finds directions that maximize class separability.

39. **What is feature scaling?**

    * The process of normalizing the range of independent variables or features of data. Examples: standardization, min-max scaling.

40. **What is cross-validation?**

    * A technique to evaluate the performance of a model by partitioning the data into subsets, training on some, and testing on others.

41. **What is the bias-variance trade-off?**

    * A central problem in supervised learning. **Bias:** Error from erroneous assumptions in the learning algorithm. **Variance:** Error from sensitivity to small fluctuations in the training set.

42. **What is regularization?**

    * Techniques used to prevent overfitting by adding a penalty term to the loss function. Examples: L1 (Lasso), L2 (Ridge).

43. **What is a generative model?**

    * A model that learns the joint probability distribution of the input features and the target variable, and can generate new data instances.

44. **What is a discriminative model?**

* A model that learns the conditional probability of the target variable given the input features, and is used for classification or regression.

45. **Difference between generative and discriminative models?**

    * Generative models learn the full data distribution, discriminative models learn the decision boundary.

46. **What is a Hidden Markov Model (HMM)?**

    * A statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.

47. **What is a Gaussian Mixture Model (GMM)?**

    * A probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.

48. **What is Expectation-Maximization (EM) algorithm?**

    * An iterative algorithm to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables.

49. **What is Optical Character Recognition (OCR)?**

    * The electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text.

50. **What is Speech Recognition?**

    * The process by which spoken words are converted to text.

51. **What is Image Segmentation?**

    * The process of partitioning a digital image into multiple segments (sets of pixels). The goal is to simplify and/or change the representation of an image into something more meaningful and easier to analyze.

52. **What is Object Detection?**

    * A computer vision technique that allows us to identify and locate objects in an image or video.

53. **What is Face Recognition?**

    * A technology capable of identifying or verifying a person from a digital image or a video frame from a video source.

54. **What is a feature vector?**

    * An n-dimensional vector of numerical features that represent some object. Used as input to machine learning algorithms.

55. **What is a distance metric?**

* A function that defines the distance between each pair of elements of a set. Examples: Euclidean distance, Manhattan distance.

56. **What is a similarity measure?**

    * A measure of how alike two data objects are. Examples: Cosine similarity, Jaccard similarity.

57. **What is a prototype?**

    * A representative example of a class or cluster.

58. **What is a template matching?**

    * A technique in digital image processing for finding small parts of an image which match a template image.

59. **What is a filter in image processing?**

    * A technique used to modify or enhance an image. Examples: Gaussian filter, Median filter.

60. **What is edge detection?**

    * An image processing technique for finding boundaries of objects within images. Examples: Sobel, Canny.

61. **What is Hough Transform?**

    * A feature extraction technique used in image analysis, computer vision, and digital image processing. It is used to detect lines, circles, or other shapes.

62. **What is SIFT (Scale-Invariant Feature Transform)?**

    * An algorithm to detect and describe local features in images. Used for object recognition.

63. **What is SURF (Speeded Up Robust Features)?**

    * A robust local feature detector and descriptor, often used in computer vision tasks like object recognition and image registration.

64. **What is Bag of Visual Words (BoVW)?**

    * A technique used in computer vision for image classification, by representing images as a collection of visual features.

65. **What is a Support Vector Regression (SVR)?**

    * An extension of Support Vector Machines (SVMs) for regression problems.

66. **What is a Gaussian Process?**

    * A stochastic process (a collection of random variables indexed by time or space) such that every finite collection of those random variables has a multivariate normal distribution.

67. **What is an Artificial Immune System (AIS)?**

    * A class of computationally intelligent systems inspired by the principles and processes of the vertebrate immune system.

68. **What is Swarm Intelligence?**

    * The collective behavior of decentralized, self-organized systems, natural or artificial. Examples: Ant Colony Optimization, Particle Swarm Optimization.

69. **What is a Genetic Algorithm?**

    * A metaheuristic inspired by the process of natural selection, belonging to the larger class of evolutionary algorithms.

70. **What is a Fuzzy Logic System?**

    * A computing approach based on 'degrees of truth' rather than the usual 'true or false' (1 or 0) Boolean logic.

Neural Networks

1. **What is a Neural Network?**
    - A computational model inspired by the structure and function of biological neural networks. It consists of interconnected nodes (neurons) organized in layers.
2. **What is a Neuron (Perceptron)?**
    - The basic building block of a neural network. It receives inputs, applies weights, sums them, adds a bias, and passes the result through an activation function.
3. **What are the components of a neuron?**
    - Inputs, Weights, Bias, Summation Function, Activation Function, Output.
4. **What is an activation function?**
    - A function that introduces non-linearity into the output of a neuron. Examples: Sigmoid, ReLU, Tanh, Softmax.
5. **Why are activation functions important?**
    - They allow neural networks to learn complex patterns and model non-linear relationships in data.
6. **Explain the Sigmoid activation function.**
    - Outputs values between 0 and 1, often used in the output layer for binary classification. Suffers from vanishing gradient problem.
7. **Explain the ReLU (Rectified Linear Unit) activation function.**

- Outputs the input directly if it is positive, otherwise outputs zero. Popular due to its computational efficiency and ability to mitigate vanishing gradient.

8. **Explain the Softmax activation function.**
    - Used in the output layer of a neural network for multi-class classification. It converts a vector of raw scores into a vector of probabilities that sum to 1.
9. **What is a feedforward neural network?**
    - A type of neural network where connections between nodes do not form a cycle. Information flows in one direction, from input to output.
10. **What is a multi-layer perceptron (MLP)?**

    * A feedforward neural network with one or more hidden layers. It can learn non-linear relationships.
11. **What is backpropagation?**

    * A supervised learning algorithm for training artificial neural networks. It calculates the gradient of the loss function with respect to the weights of the network, allowing for weight updates.
12. **What is a loss function (cost function)?**

    * A function that quantifies the difference between the predicted output of the model and the actual target value. The goal of training is to minimize this function.
13. **Give examples of loss functions.**

    * Mean Squared Error (MSE) for regression, Cross-Entropy for classification.
14. **What is an optimizer?**

    * An algorithm or function that modifies the attributes of the neural network, such as weights and learning rate, in order to reduce the loss function.
15. **Give examples of optimizers.**

    * Gradient Descent, Stochastic Gradient Descent (SGD), Adam, RMSprop.
16. **Explain Gradient Descent.**

    * An iterative optimization algorithm for finding the minimum of a function. It takes steps proportional to the negative of the gradient of the function at the current point.
17. **What is learning rate?**

    * A hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated.
18. **What is overfitting in neural networks?**

* When the model performs well on the training data but poorly on unseen data, often due to learning noise in the training data.

19. **How to prevent overfitting in neural networks?**

    * Regularization (L1, L2), Dropout, Early Stopping, Data Augmentation.

20. **What is Dropout?**

    * A regularization technique where randomly selected neurons are ignored during training. This prevents complex co-adaptations on training data.

21. **What is early stopping?**

    * A regularization technique where training is stopped when the performance on a validation set starts to degrade, even if the performance on the training set is still improving.

22. **What is regularization (L1, L2)?**

    * Techniques that add a penalty to the loss function to discourage large weights, reducing overfitting.

23. **What is a hyperparameter?**

    * A parameter whose value is set before the learning process begins. Examples: learning rate, number of hidden layers, number of neurons.

24. **What is a Convolutional Neural Network (CNN)?**

    * A class of deep neural networks, most commonly applied to analyzing visual imagery. It uses convolutional layers to automatically and adaptively learn spatial hierarchies of features.

25. **What are the main layers in a CNN?**

    * Convolutional Layer, Pooling Layer, Fully Connected Layer.

26. **Explain the Convolutional Layer.**

    * Applies a convolution operation to the input, passing the result to the next layer. It uses filters (kernels) to detect features.

27. **Explain the Pooling Layer.**

    * Reduces the spatial dimensions (width and height) of the input volume for the next convolutional layer. Examples: Max Pooling, Average Pooling.

28. **What is a Recurrent Neural Network (RNN)?**

    * A class of neural networks where connections between nodes form a directed graph along a temporal sequence. This allows them to exhibit temporal dynamic behavior.

29. **What is the vanishing gradient problem?**

* A problem in training RNNs and deep feedforward networks where the gradients become extremely small as they propagate backward through many layers, making it difficult to learn long-range dependencies.

30. **What is the exploding gradient problem?**

    * A problem in training RNNs where the gradients become extremely large, leading to unstable training and large weight updates.

31. **How to mitigate vanishing/exploding gradients?**

    * **Vanishing:** ReLU activation, LSTM/GRU, skip connections. **Exploding:** Gradient clipping.

32. **What is an LSTM (Long Short-Term Memory) network?**

    * A type of RNN capable of learning long-term dependencies. It has memory cells and gates (input, forget, output) to control information flow.

33. **What is a GRU (Gated Recurrent Unit)?**

    * A simpler variant of LSTM, also designed to handle vanishing gradient problem in RNNs. It has fewer gates (reset, update) than LSTM.

34. **What is an Autoencoder?**

    * A type of artificial neural network used for learning efficient data codings (representations) in an unsupervised manner. It has an encoder that compresses input into a latent-space representation, and a decoder that reconstructs the input from the latent space.

35. **What is a Generative Adversarial Network (GAN)?**

    * A class of machine learning frameworks where two neural networks (a generator and a discriminator) contest with each other in a zero-sum game.

36. **Explain the Generator and Discriminator in GANs.**

    * **Generator:** Creates new data instances that resemble the training data.
    Discriminator: Tries to distinguish between real and fake data.

37. **What is Transfer Learning?**

    * A machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

38. **What is fine-tuning?**

    * A technique in transfer learning where a pre-trained model is further trained on a new, smaller dataset to adapt it to a specific task.

39. **What is batch normalization?**

* A technique for improving the speed, performance, and stability of artificial neural networks by normalizing the inputs to layers.

40. **What is a tensor?**

    * A multi-dimensional array. In neural networks, data is typically represented as tensors.

41. **What is a computational graph?**

    * A way to represent mathematical equations or algorithms as a graph. Used in deep learning frameworks to define and execute models.

42. **What is an embedding?**

    * A low-dimensional, learned representation of discrete variables (e.g., words, categories) that captures semantic relationships.

43. **What is word embedding?**

    * A representation of words in a vector space where words with similar meanings have similar vector representations.

44. **What is a Transformer?**

    * A deep learning model introduced in 2017, primarily used in natural language processing (NLP). It relies entirely on self-attention mechanisms.

45. **What is self-attention?**

    * A mechanism that allows the model to weigh the importance of different words in the input sequence when processing each word.

46. **What is a positional encoding?**

    * A technique used in Transformers to inject information about the relative or absolute position of tokens in the sequence.

47. **What is a sequence-to-sequence model?**

    * A model that takes a sequence as input and outputs another sequence. Often used in machine translation.

48. **What is an encoder-decoder architecture?**

    * A common architecture for sequence-to-sequence models, where an encoder maps the input sequence to a context vector, and a decoder generates the output sequence from the context vector.

49. **What is a residual connection (skip connection)?**

    * A connection that bypasses one or more layers in a neural network, allowing gradients to flow more easily and helping to train very deep networks.

50. **What is a bottleneck layer?**

* A layer that reduces the dimensionality of the input, often used in autoencoders or in deep networks to reduce computational cost.

51. **What is a generative model vs. discriminative model in NN?**

    * **Generative:** Learns the distribution of data to generate new samples. **Discriminative:** Learns to distinguish between different classes.

52. **What is a capsule network?**

    * A type of neural network that aims to address some limitations of CNNs, particularly regarding spatial hierarchies and viewpoint invariance.

53. **What is a Graph Neural Network (GNN)?**

    * A type of neural network designed to operate on graph-structured data.

54. **What is a Spiking Neural Network (SNN)?**

    * A type of neural network that more closely mimics biological neural networks by using discrete 'spikes' to transmit information.

55. **What is Quantization in NN?**

    * The process of reducing the precision of numbers used to represent a model's parameters and activations, typically to reduce model size and speed up inference.

56. **What is Pruning in NN?**

    * A technique to reduce the size of a neural network by removing redundant or less important connections (weights).

57. **What is Knowledge Distillation?**

    * A technique where a smaller, simpler 'student' model is trained to mimic the behavior of a larger, more complex 'teacher' model.

58. **What is Adversarial Training?**

    * A technique to improve the robustness of a model by training it on adversarial examples (inputs intentionally perturbed to fool the model).

59. **What is Explainable AI (XAI)?**

    * An emerging field of machine learning that aims to make AI models more transparent and understandable to humans.

60. **What is Federated Learning?**

    * A machine learning approach that trains an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging their data samples.

61. **What is Reinforcement Learning (RL)?**

* A type of machine learning where an agent learns to make decisions by performing actions in an environment to maximize a cumulative reward.

62. **What is an Agent, Environment, State, Action, Reward in RL?**

    * **Agent:** The learner/decision-maker. **Environment:** The world the agent interacts with. **State:** Current situation of the agent. **Action:** Agent's move. **Reward:** Feedback from the environment.

63. **What is a Policy in RL?**

    * A mapping from states to actions, defining the agent's behavior.

64. **What is a Value Function in RL?**

    * A function that estimates how good it is for the agent to be in a given state or to perform a given action in a given state.

65. **What is Q-learning?**

    * A model-free reinforcement learning algorithm to learn the value of an action in a particular state.

66. **What is Deep Q-Network (DQN)?**

    * A type of Q-learning that uses a deep neural network to approximate the Q-value function.

67. **What is Policy Gradient?**

    * A class of reinforcement learning algorithms that directly learn a policy function that maps states to actions.

68. **What is Actor-Critic method?**

    * A reinforcement learning method that combines both policy gradient (actor) and value-based (critic) approaches.

Artificial Intelligence

1. **What is Artificial Intelligence (AI)?**
    - The simulation of human intelligence in machines that are programmed to think like humans and mimic their actions.
2. **What are the main goals of AI?**
    - Reasoning, Knowledge Representation, Planning, Learning, Natural Language Processing (NLP), Perception, Motion and Manipulation.
3. **What is the Turing Test?**
    - A test of a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human.

4. **What is Strong AI vs. Weak AI?**
    - **Strong AI (Artificial General Intelligence - AGI):** A machine with intellect capable of performing any intellectual task that a human being can. **Weak AI (Artificial Narrow Intelligence - ANI):** AI designed and trained for a particular task.
5. **What is Machine Learning (ML)?**
    - A subset of AI that enables systems to learn from data, identify patterns, and make decisions with minimal human intervention.
6. **What is Deep Learning (DL)?**
    - A subset of Machine Learning that uses artificial neural networks with multiple layers to learn from vast amounts of data.
7. **Difference between AI, ML, and DL?**
    - AI is the broad concept of machines mimicking human intelligence. ML is a subset of AI that focuses on learning from data. DL is a subset of ML that uses deep neural networks.
8. **What is an Agent in AI?**
    - Anything that can perceive its environment through sensors and act upon that environment through actuators.
9. **What are the types of AI agents?**
    - Simple reflex agents, Model-based reflex agents, Goal-based agents, Utility-based agents, Learning agents.
10. **What is a search algorithm in AI?**

    * Algorithms used to find a solution path from an initial state to a goal state in a problem space.
11. **Explain uninformed search strategies.**

    * Search strategies that do not use any domain knowledge to guide the search. Examples: BFS, DFS, Uniform Cost Search.
12. **Explain informed search strategies.**

    * Search strategies that use heuristic functions to guide the search towards the goal. Examples: Greedy Best-First Search, A* Search.
13. **What is a heuristic function?**

    * A function that estimates how close a state is to the goal state. Used to guide informed search algorithms.
14. *Explain A search algorithm.**

* An informed search algorithm that finds the shortest path between a starting node and a goal node. It uses a heuristic function and the cost to reach the current node.

15. **What is minimax algorithm?**

    * A decision rule used in game theory, decision theory, statistics, and philosophy for minimizing the possible loss for a worst-case (maximum loss) scenario.

16. **What is alpha-beta pruning?**

    * An optimization technique for minimax algorithm that reduces the number of nodes evaluated in the search tree.

17. **What is knowledge representation?**

    * The field of AI dedicated to representing information about the world in a form that a computer system can use to solve complex tasks.

18. **What are common knowledge representation techniques?**

    * Predicate Logic, Semantic Networks, Frames, Scripts, Ontologies.

19. **What is Natural Language Processing (NLP)?**

    * A field of AI that enables computers to understand, interpret, and generate human language.

20. **What are common NLP tasks?**

    * Tokenization, Part-of-Speech Tagging, Named Entity Recognition, Sentiment Analysis, Machine Translation.

21. **What is tokenization?**

    * The process of breaking a stream of text into words, phrases, symbols, or other meaningful elements called tokens.

22. **What is stemming and lemmatization?**

    * **Stemming:** Reducing words to their root form (e.g., 'running' -> 'run').
    Lemmatization: Reducing words to their base or dictionary form (e.g., 'better' -> 'good').

23. **What is sentiment analysis?**

    * The process of determining the emotional tone behind a body of text.

24. **What is machine translation?**

    * The task of automatically translating text or speech from one natural language to another.

25. **What is computer vision?**

    * A field of AI that enables computers to 'see' and interpret visual information from the world.

26. **What are common computer vision tasks?**

    * Image Classification, Object Detection, Image Segmentation, Face Recognition.

27. **What is planning in AI?**

    * The process of finding a sequence of actions that will achieve a goal state from an initial state.

28. **What is an expert system?**

    * A computer system that emulates the decision-making ability of a human expert.

29. **What is a fuzzy logic system?**

    * A computing approach based on 'degrees of truth' rather than the usual 'true or false' (1 or 0) Boolean logic.

30. **What is an Artificial Neural Network (ANN)?**

    * A computational model inspired by the structure and function of biological neural networks.

31. **What is Reinforcement Learning (RL)?**

    * A type of machine learning where an agent learns to make decisions by performing actions in an environment to maximize a cumulative reward.

32. **What is an Agent, Environment, State, Action, Reward in RL?**

    * **Agent:** The learner/decision-maker. **Environment:** The world the agent interacts with. **State:** Current situation of the agent. **Action:** Agent's move. **Reward:** Feedback from the environment.

33. **What is a Policy in RL?**

    * A mapping from states to actions, defining the agent's behavior.

34. **What is a Value Function in RL?**

    * A function that estimates how good it is for the agent to be in a given state or to perform a given action in a given state.

35. **What is Q-learning?**

    * A model-free reinforcement learning algorithm to learn the value of an action in a particular state.

36. **What is Deep Q-Network (DQN)?**

    * A type of Q-learning that uses a deep neural network to approximate the Q-value function.

37. **What is Policy Gradient?**

* A class of reinforcement learning algorithms that directly learn a policy function that maps states to actions.

38. **What is Actor-Critic method?**

    * A reinforcement learning method that combines both policy gradient (actor) and value-based (critic) approaches.

39. **What is Genetic Algorithm?**

    * A metaheuristic inspired by the process of natural selection, belonging to the larger class of evolutionary algorithms.

40. **What is Swarm Intelligence?**

    * The collective behavior of decentralized, self-organized systems, natural or artificial. Examples: Ant Colony Optimization, Particle Swarm Optimization.

41. **What is an Ontology in AI?**

    * A formal representation of knowledge as a set of concepts within a domain and the relationships between those concepts.

42. **What is a Semantic Network?**

    * A graph-based knowledge representation technique where nodes represent concepts and links represent relationships between them.

43. **What is a Frame in AI?**

    * A data structure for representing a stereotyped situation, like a restaurant script. It has slots for various aspects of the situation.

44. **What is an Inference Engine?**

    * A component of an expert system that applies logical rules to the knowledge base to deduce new facts or answer queries.

45. **What is a Knowledge Base?**

    * A special kind of database for knowledge management, providing the means for the computerized collection, organization, and retrieval of knowledge.

46. **What is an AI chatbot?**

    * A computer program designed to simulate human conversation through voice commands or text chats or both.

47. **What is Robotic Process Automation (RPA)?**

    * A technology that allows anyone to configure computer software, or a 'robot,' to emulate and integrate the actions of a human interacting within digital systems to execute a business process.

48. **What is Cognitive Computing?**

* A technology platform that combines artificial intelligence and machine learning with other advanced technologies to enable human-like interaction with systems.

49. **What is AI Ethics?**

    * A field of study concerned with the moral implications of designing, developing, and deploying artificial intelligence.

50. **What is Explainable AI (XAI)?**

    * An emerging field of machine learning that aims to make AI models more transparent and understandable to humans.

51. **What is AI Bias?**

    * Systematic and repeatable errors in a computer system that create unfair outcomes, such as favoring one group over others.

52. **What is AI Safety?**

    * A field of research focused on ensuring that advanced AI systems are beneficial and do not cause unintended harm.

53. **What is Transfer Learning in AI?**

    * A machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

54. **What is Federated Learning in AI?**

    * A machine learning approach that trains an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging their data samples.

55. **What is a Bayesian Network?**

    * A probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph.

56. **What is Markov Decision Process (MDP)?**

    * A mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker.

57. **What is a Utility Function in AI?**

    * A function that maps states or outcomes to real numbers, representing the desirability of those states or outcomes.

58. **What is a Game Tree?**

    * A directed graph whose nodes are positions in a game and whose edges are moves. Used in game theory to analyze possible moves.

59. **What is a Constraint Satisfaction Problem (CSP)?**

* A problem defined by a set of variables, each with a domain of possible values, and a set of constraints that restrict the values that the variables can simultaneously take.

60. **What is a Genetic Programming?**

    * A type of evolutionary algorithm that automatically generates computer programs to solve a problem.

61. **What is Multi-Agent System (MAS)?**

    * A computerized system composed of multiple interacting intelligent agents.

62. **What is Human-in-the-Loop (HITL) AI?**

    * An approach to AI where humans are involved in the loop of machine learning model development and deployment.

63. **What is Cognitive Robotics?**

    * A field of robotics that studies how to build robots that can reason, learn, and interact with humans in a natural way.

64. **What is Explainable AI (XAI)?**

    * An emerging field of machine learning that aims to make AI models more transparent and understandable to humans.

Deep Learning

1. **What is Deep Learning?**
    - A subset of Machine Learning that uses artificial neural networks with multiple layers (deep neural networks) to learn from vast amounts of data.
2. **How is Deep Learning different from traditional Machine Learning?**
    - DL automatically learns features from data, while traditional ML often requires manual feature engineering. DL excels with large datasets and complex patterns.
3. **What is a deep neural network?**
    - A neural network with multiple hidden layers, allowing it to learn hierarchical representations of data.
4. **What is a Convolutional Neural Network (CNN)?**
    - A class of deep neural networks, most commonly applied to analyzing visual imagery. It uses convolutional layers to automatically and adaptively learn spatial hierarchies of features.
5. **What are the main layers in a CNN?**
    - Convolutional Layer, Pooling Layer, Fully Connected Layer.

6. **Explain the Convolutional Layer.**
    - Applies a convolution operation to the input, passing the result to the next layer. It uses filters (kernels) to detect features.
7. **Explain the Pooling Layer.**
    - Reduces the spatial dimensions (width and height) of the input volume for the next convolutional layer. Examples: Max Pooling, Average Pooling.
8. **What is a Recurrent Neural Network (RNN)?**
    - A class of neural networks where connections between nodes form a directed graph along a temporal sequence. This allows them to exhibit temporal dynamic behavior.
9. **What is the vanishing gradient problem?**
    - A problem in training RNNs and deep feedforward networks where the gradients become extremely small as they propagate backward through many layers, making it difficult to learn long-range dependencies.
10. **What is the exploding gradient problem?**

    * A problem in training RNNs where the gradients become extremely large, leading to unstable training and large weight updates.
11. **How to mitigate vanishing/exploding gradients?**

    * **Vanishing:** ReLU activation, LSTM/GRU, skip connections. **Exploding:** Gradient clipping.
12. **What is an LSTM (Long Short-Term Memory) network?**

    * A type of RNN capable of learning long-term dependencies. It has memory cells and gates (input, forget, output) to control information flow.
13. **What is a GRU (Gated Recurrent Unit)?**

    * A simpler variant of LSTM, also designed to handle vanishing gradient problem in RNNs. It has fewer gates (reset, update) than LSTM.
14. **What is an Autoencoder?**

    * A type of artificial neural network used for learning efficient data codings (representations) in an unsupervised manner. It has an encoder that compresses input into a latent-space representation, and a decoder that reconstructs the input from the latent space.
15. **What is a Generative Adversarial Network (GAN)?**

    * A class of machine learning frameworks where two neural networks (a generator and a discriminator) contest with each other in a zero-sum game.
16. **Explain the Generator and Discriminator in GANs.**

* **Generator:** Creates new data instances that resemble the training data.
Discriminator: Tries to distinguish between real and fake data.

17. **What is Transfer Learning?**

    * A machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

18. **What is fine-tuning?**

    * A technique in transfer learning where a pre-trained model is further trained on a new, smaller dataset to adapt it to a specific task.

19. **What is batch normalization?**

    * A technique for improving the speed, performance, and stability of artificial neural networks by normalizing the inputs to layers.

20. **What is a tensor?**

    * A multi-dimensional array. In neural networks, data is typically represented as tensors.

21. **What is a computational graph?**

    * A way to represent mathematical equations or algorithms as a graph. Used in deep learning frameworks to define and execute models.

22. **What is an embedding?**

    * A low-dimensional, learned representation of discrete variables (e.g., words, categories) that captures semantic relationships.

23. **What is word embedding?**

    * A representation of words in a vector space where words with similar meanings have similar vector representations.

24. **What is a Transformer?**

    * A deep learning model introduced in 2017, primarily used in natural language processing (NLP). It relies entirely on self-attention mechanisms.

25. **What is self-attention?**

    * A mechanism that allows the model to weigh the importance of different words in the input sequence when processing each word.

26. **What is a positional encoding?**

    * A technique used in Transformers to inject information about the relative or absolute position of tokens in the sequence.

27. **What is a sequence-to-sequence model?**

    * A model that takes a sequence as input and outputs another sequence. Often used in machine translation.

28. **What is an encoder-decoder architecture?**

    * A common architecture for sequence-to-sequence models, where an encoder maps the input sequence to a context vector, and a decoder generates the output sequence from the context vector.

29. **What is a residual connection (skip connection)?**

    * A connection that bypasses one or more layers in a neural network, allowing gradients to flow more easily and helping to train very deep networks.

30. **What is a bottleneck layer?**

    * A layer that reduces the dimensionality of the input, often used in autoencoders or in deep networks to reduce computational cost.

31. **What is a generative model vs. discriminative model in DL?**

    * **Generative:** Learns the distribution of data to generate new samples. **Discriminative:** Learns to distinguish between different classes.

32. **What is a capsule network?**

    * A type of neural network that aims to address some limitations of CNNs, particularly regarding spatial hierarchies and viewpoint invariance.

33. **What is a Graph Neural Network (GNN)?**

    * A type of neural network designed to operate on graph-structured data.

34. **What is a Spiking Neural Network (SNN)?**

    * A type of neural network that more closely mimics biological neural networks by using discrete 'spikes' to transmit information.

35. **What is Quantization in DL?**

    * The process of reducing the precision of numbers used to represent a model's parameters and activations, typically to reduce model size and speed up inference.

36. **What is Pruning in DL?**

    * A technique to reduce the size of a neural network by removing redundant or less important connections (weights).

37. **What is Knowledge Distillation?**

    * A technique where a smaller, simpler 'student' model is trained to mimic the behavior of a larger, more complex 'teacher' model.

38. **What is Adversarial Training?**

    * A technique to improve the robustness of a model by training it on adversarial examples (inputs intentionally perturbed to fool the model).

39. **What is Explainable AI (XAI) in DL?**

* An emerging field of machine learning that aims to make AI models more transparent and understandable to humans.

40. **What is Federated Learning in DL?**

    * A machine learning approach that trains an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging their data samples.

41. **What is Reinforcement Learning (RL) in DL?**

    * A type of machine learning where an agent learns to make decisions by performing actions in an environment to maximize a cumulative reward.

42. **What is a Policy in RL?**

    * A mapping from states to actions, defining the agent's behavior.

43. **What is a Value Function in RL?**

    * A function that estimates how good it is for the agent to be in a given state or to perform a given action in a given state.

44. **What is Q-learning?**

    * A model-free reinforcement learning algorithm to learn the value of an action in a particular state.

45. **What is Deep Q-Network (DQN)?**

    * A type of Q-learning that uses a deep neural network to approximate the Q-value function.

46. **What is Policy Gradient?**

    * A class of reinforcement learning algorithms that directly learn a policy function that maps states to actions.

47. **What is Actor-Critic method?**

    * A reinforcement learning method that combines both policy gradient (actor) and value-based (critic) approaches.

48. **What is a Generative Pre-trained Transformer (GPT)?**

    * A type of large language model (LLM) that uses the Transformer architecture and is pre-trained on a massive corpus of text data.

49. **What is BERT (Bidirectional Encoder Representations from Transformers)?**

    * A Transformer-based machine learning technique for natural language processing (NLP) pre-training.

50. **What is an attention mechanism?**

* A component in neural networks that allows the model to focus on specific parts of the input sequence when making predictions.

51. **What is a Vision Transformer (ViT)?**

    * A Transformer model applied directly to sequences of image patches for image classification.

52. **What is a Diffusion Model?**

    * A generative model that learns to denoise data by gradually adding noise and then reversing the process to generate new data.

53. **What is a Variational Autoencoder (VAE)?**

    * A generative model that provides a probabilistic way of describing an observation in latent space.

54. **What is a Recurrent Convolutional Neural Network (RCNN)?**

    * A combination of RNNs and CNNs, often used for tasks involving both spatial and temporal dependencies.

55. **What is a Long Short-Term Recurrent Convolutional Network (LRCN)?**

    * A type of deep neural network that combines the strengths of CNNs for spatial feature extraction and LSTMs for temporal sequence modeling.

56. **What is a Deep Belief Network (DBN)?**

    * A generative graphical model, or a class of deep neural networks, composed of multiple layers of hidden units, where each layer is connected to both the previous and next layers.

57. **What is a Restricted Boltzmann Machine (RBM)?**

    * A generative stochastic artificial neural network that can learn a probability distribution over its set of inputs.

58. **What is a Boltzmann Machine?**

    * A type of stochastic recurrent neural network that can be used to solve optimization problems and learn complex patterns.

59. **What is a Hopfield Network?**

    * A type of recurrent artificial neural network that serves as content-addressable memory systems with binary units.

60. **What is a Self-Organizing Map (SOM)?**

    * A type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional, discretized representation of the input space of the training samples, called a map.

61. **What is a Radial Basis Function Network (RBFN)?**

    * A type of artificial neural network that uses radial basis functions as activation functions. They are typically used for function approximation, time series prediction, and classification.

62. **What is a Extreme Learning Machine (ELM)?**

    * A feedforward neural network for classification, regression, clustering, sparse approximation, compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes (not just the weights connecting hidden nodes to output nodes) are randomly chosen and need not be tuned.

63. **What is a Liquid State Machine (LSM)?**

    * A type of recurrent neural network that maps an input into a higher-dimensional computational space.

64. **What is an Echo State Network (ESN)?**

    * A type of recurrent neural network with a sparsely connected random hidden layer (called a reservoir) and a learnable output layer.

65. **What is a Deep Reinforcement Learning (DRL)?**

    * The combination of deep learning and reinforcement learning. It uses deep neural networks to learn policies or value functions in RL.

66. **What is an AlphaGo/AlphaZero?**

    * AI programs developed by DeepMind that use DRL and tree search (Monte Carlo Tree Search) to achieve superhuman performance in games like Go, Chess, and Shogi.

67. **What is a Generative AI?**

    * A type of artificial intelligence that can create new content, such as images, text, audio, and video, that is similar to human-created content.

68. **What is a Large Language Model (LLM)?**

    * A type of AI model that is trained on a massive amount of text data to understand, generate, and respond to human language.

69. **What is a Foundation Model?**

    * A large AI model trained on a vast quantity of data at scale such that it can be adapted to a wide range of downstream tasks.

70. **What is Few-shot Learning?**

    * A machine learning setting where a model is trained on a very small number of examples for a new task.

71. **What is Zero-shot Learning?**

* A machine learning setting where a model is trained to solve a task without seeing any examples of that task during training.

72. **What is Prompt Engineering?**

    * The process of designing and refining inputs (prompts) for AI models, especially large language models, to guide them toward generating desired outputs.

73. **What is Retrieval-Augmented Generation (RAG)?**

    * A technique that enhances the output of large language models by retrieving relevant information from an external knowledge base before generating a response.

74. **What is Fine-tuning in LLMs?**

    * The process of further training a pre-trained large language model on a smaller, task-specific dataset to adapt it to a particular use case.

75. **What is LoRA (Low-Rank Adaptation)?**

    * A parameter-efficient fine-tuning technique for large language models that reduces the number of trainable parameters by injecting small, low-rank matrices into the model.

76. **What is Quantization in LLMs?**

    * The process of reducing the precision of the numerical representations of weights and activations in large language models, typically to reduce memory footprint and speed up inference.

77. **What is a Mixture of Experts (MoE) model?**

    * A type of neural network architecture that consists of multiple 'expert' sub-networks, with a 'gate' network that learns to select which expert to use for a given input.

78. **What is a Multimodal AI?**

    * AI systems that can process and understand information from multiple modalities, such as text, images, audio, and video.

79. **What is Federated Learning in Deep Learning?**

    * A machine learning approach that trains an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging their data samples.

80. **What is On-Device AI?**

    * AI models that run directly on edge devices (e.g., smartphones, IoT devices) rather than in the cloud.

81. **What is Edge AI?**

* The deployment of AI algorithms on edge devices rather than in a centralized cloud computing system.

82. **What is Neuromorphic Computing?**

    * A new computing paradigm that aims to mimic the brain's structure and function to achieve energy-efficient and high-performance AI.

83. **What is Quantum Machine Learning?**

    * An interdisciplinary field that explores the interplay between quantum computing and machine learning.

Cyber Security

1. **What is Cyber Security?**
    - The practice of protecting systems, networks, and programs from digital attacks.
2. **What are the three pillars of cybersecurity (CIA Triad)?**
    - **Confidentiality:** Protecting information from unauthorized access. **Integrity:** Ensuring information is accurate and trustworthy. **Availability:** Ensuring authorized users have access to information and systems when needed.
3. **What is a threat, vulnerability, and risk?**
    - **Threat:** A potential danger that might exploit a vulnerability. **Vulnerability:** A weakness in a system that can be exploited. **Risk:** The potential for loss or damage when a threat exploits a vulnerability.
4. **What is a firewall?**
    - A network security device that monitors and filters incoming and outgoing network traffic based on an organization's previously established security policies.
5. **What is the difference between a stateful and stateless firewall?**
    - **Stateless:** Filters packets based on individual packet headers. **Stateful:** Monitors the state of active connections and makes filtering decisions based on the context of the traffic.
6. **What is an IDS (Intrusion Detection System)?**
    - A system that monitors network traffic for suspicious activity and alerts when such activity is found.
7. **What is an IPS (Intrusion Prevention System)?**
    - A system that monitors network traffic for suspicious activity and automatically takes action to prevent it.
8. **Difference between IDS and IPS?**

- IDS detects and alerts, IPS detects and prevents.

9. **What is a VPN (Virtual Private Network)?**
    - A technology that creates a secure, encrypted connection over a less secure network, such as the internet.
10. **What is encryption?**

    * The process of converting information or data into a code to prevent unauthorized access.
11. **What is decryption?**

    * The process of converting encrypted data back into its original form.
12. **What is symmetric encryption?**

    * Uses a single key for both encryption and decryption. Examples: AES, DES.
13. **What is asymmetric encryption (public-key cryptography)?**

    * Uses a pair of keys: a public key for encryption and a private key for decryption. Examples: RSA, ECC.
14. **What is a digital signature?**

    * A mathematical scheme for demonstrating the authenticity of digital messages or documents. It ensures integrity and non-repudiation.
15. **What is a hash function in cybersecurity?**

    * A mathematical algorithm that maps data of arbitrary size to a fixed-size bit array. Used for data integrity checks.
16. **What is a collision in hashing?**

    * When two different inputs produce the same hash output.
17. **What is a brute-force attack?**

    * A trial-and-error method used to obtain information such as user passwords or personal identification numbers (PINs). It tries every possible combination.
18. **What is a dictionary attack?**

    * A type of brute-force attack that attempts to defeat a password authentication mechanism by trying to determine its decryption key or passphrase by trying hundreds or sometimes millions of likely possibilities, such as words in a dictionary.
19. **What is phishing?**

    * A type of social engineering attack often used to steal user data, including login credentials and credit card numbers.
20. **What is spear phishing?**

* A targeted phishing attempt, often appearing to come from a trusted source.

21. **What is ransomware?**

    * A type of malicious software that encrypts a victim's files and demands a ransom payment to decrypt them.

22. **What is malware?**

    * Malicious software, including viruses, worms, Trojans, ransomware, spyware, adware, etc.

23. **What is a virus?**

    * A type of malware that, when executed, replicates itself by modifying other computer programs and inserting its own code.

24. **What is a worm?**

    * A standalone malware computer program that replicates itself to spread to other computers.

25. **What is a Trojan horse?**

    * A type of malware that disguises itself as legitimate software but carries out malicious functions.

26. **What is a rootkit?**

    * A collection of computer software, typically malicious, designed to enable access to a computer or an area of its software that is not otherwise allowed and often masks its existence or the existence of other software.

27. **What is a botnet?**

    * A network of private computers infected with malicious software and controlled as a group without the owners' knowledge, e.g., to send spam messages.

28. **What is a DDoS (Distributed Denial of Service) attack?**

    * A malicious attempt to disrupt normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

29. **What is SQL Injection?**

    * A code injection technique used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution.

30. **What is XSS (Cross-Site Scripting)?**

    * A type of security vulnerability typically found in web applications. XSS enables attackers to inject client-side scripts into web pages viewed by other users.

31. **What is CSRF (Cross-Site Request Forgery)?**

* An attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.

32. **What is a Zero-day exploit?**

    * A cyberattack that exploits a previously unknown vulnerability in a computer application or operating system.

33. **What is social engineering?**

    * The psychological manipulation of people into performing actions or divulging confidential information.

34. **What is penetration testing (pen testing)?**

    * An authorized simulated cyberattack on a computer system, performed to evaluate the security of the system.

35. **What is vulnerability scanning?**

    * An automated process of identifying security weaknesses in a network or system.

36. **Difference between penetration testing and vulnerability scanning?**

    * Vulnerability scanning identifies known weaknesses, pen testing exploits them to assess actual risk.

37. **What is a security policy?**

    * A document that outlines the rules and procedures for protecting an organization's information assets.

38. **What is incident response?**

    * An organized approach to addressing and managing the aftermath of a security breach or cyberattack.

39. **What is SIEM (Security Information and Event Management)?**

    * A solution that helps organizations detect, analyze, and respond to security threats by collecting and analyzing security event data.

40. **What is SOC (Security Operations Center)?**

    * A centralized function within an organization that employs people, processes, and technology to continuously monitor and improve an organization's security posture.

41. **What is a security audit?**

    * A systematic evaluation of the security of a company's information system by measuring how well it conforms to a set of established criteria.

42. **What is multi-factor authentication (MFA)?**

    * A security system that requires more than one method of authentication from independent categories of credentials to verify the user's identity.

43. **What is a security token?**

    * A physical device or software-based mechanism used to authenticate a user's identity.

44. **What is a digital certificate?**

    * An electronic document used to prove the ownership of a public key. It is issued by a Certificate Authority (CA).

45. **What is PKI (Public Key Infrastructure)?**

    * A set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

46. **What is a honeypot?**

    * A security mechanism that is set up to attract and trap cyberattackers, thereby diverting them from legitimate systems.

47. **What is a sandbox in cybersecurity?**

    * An isolated testing environment that enables users to run programs or open files without affecting the application, system or platform on which they run.

48. **What is data loss prevention (DLP)?**

    * A set of tools and processes used to ensure that sensitive data is not lost, misused, or accessed by unauthorized users.

49. **What is endpoint security?**

    * The practice of securing endpoints (e.g., laptops, desktops, mobile devices) from cyber threats.

50. **What is cloud security?**

    * A set of policies, controls, procedures, and technologies that work together to protect cloud-based systems, data, and infrastructure.

51. **What is IoT security?**

    * The protection of connected devices and networks in the Internet of Things ecosystem.

52. **What is OT security (Operational Technology)?**

    * The security of industrial control systems (ICS) and other operational technology environments.

53. **What is GRC (Governance, Risk, and Compliance)?**

    * A strategy for managing an organization's overall governance, enterprise risk management, and compliance with regulations.

54. **What is GDPR (General Data Protection Regulation)?**

* A regulation in EU law on data protection and privacy in the European Union and the European Economic Area.

55. **What is HIPAA (Health Insurance Portability and Accountability Act)?**

    * A US law designed to provide privacy standards to protect patients' medical records and other health information.

56. **What is PCI DSS (Payment Card Industry Data Security Standard)?**

    * A set of security standards designed to ensure that all companies that accept, process, store or transmit credit card information maintain a secure environment.

57. **What is ISO 27001?**

    * An international standard for information security management systems (ISMS).

58. **What is a security incident?**

    * An event that violates an organization's security policies or poses a threat to its information assets.

59. **What is a security breach?**

    * A security incident in which data is stolen or taken from a system without the knowledge or authorization of the system's owner.

60. **What is a vulnerability assessment?**

    * The process of identifying, quantifying, and prioritizing the vulnerabilities in a system.

61. **What is a security framework?**

    * A structured set of guidelines and best practices to manage and reduce cybersecurity risks.

62. **What is a security control?**

    * A safeguard or countermeasure to avoid, detect, counteract, or minimize security risks to physical property, information, computer systems, or other assets.

63. **What is threat intelligence?**

    * Evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets.

64. **What is ethical hacking?**

    * The practice of using hacking skills to identify vulnerabilities in systems with the permission of the owner, to improve security.

65. **What is a bug bounty program?**

    * A program offered by many websites and software developers whereby individuals can receive recognition and compensation for reporting bugs, especially those pertaining to exploits and vulnerabilities.

66. **What is Zero Trust security?**

    * A security model that assumes no user or device can be trusted by default, even if they are inside the network perimeter.

67. **What is a Security Information and Event Management (SIEM) system?**

    * A solution that helps organizations detect, analyze, and respond to security threats by collecting and analyzing security event data.

68. **What is a Security Orchestration, Automation, and Response (SOAR) platform?**

    * A solution that helps organizations automate and orchestrate security operations tasks, such as incident response and threat hunting.

69. **What is a Security Operations Center (SOC)?**

    * A centralized function within an organization that employs people, processes, and technology to continuously monitor and improve an organization's security posture.

70. **What is a Red Team vs. Blue Team?**

    * **Red Team:** Simulates attacks against an organization's security controls. **Blue Team:** Defends against attacks and improves security posture.

71. **What is Purple Teaming?**

    * A collaborative approach where Red and Blue Teams work together to improve security.

DLD (Digital Logic Design)

1. **What is Digital Logic Design (DLD)?**
    - The process of designing digital circuits, which are the fundamental building blocks of computers and other digital systems.
2. **What is a logic gate?**
    - An elementary building block of a digital circuit. Most logic gates have two inputs and one output. Examples: AND, OR, NOT, NAND, NOR, XOR, XNOR.
3. **Explain AND gate.**
    - Output is HIGH only if all inputs are HIGH.
4. **Explain OR gate.**
    - Output is HIGH if any input is HIGH.
5. **Explain NOT gate (Inverter).**
    - Output is the inverse of the input.
6. **Explain NAND gate.**

- Output is LOW only if all inputs are HIGH (NOT AND).

7. **Explain NOR gate.**
   - Output is HIGH only if all inputs are LOW (NOT OR).
8. **Explain XOR gate (Exclusive OR).**
   - Output is HIGH if inputs are different.
9. **Explain XNOR gate (Exclusive NOR).**
   - Output is HIGH if inputs are the same.
10. **What are universal gates?**

    * Gates that can be used to implement any Boolean function. Examples: NAND, NOR.
11. **What is Boolean algebra?**

    * A branch of algebra in which the values of the variables are the truth values true and false, usually denoted 1 and 0 respectively.
12. **State De Morgan's Theorems.**

    * 1. `(A + B)' = A' * B'` (NOT (A OR B) = (NOT A) AND (NOT B)). 2. `(A * B)' = A' + B'` (NOT (A AND B) = (NOT A) OR (NOT B)).
13. **What is a Karnaugh Map (K-Map)?**

    * A graphical method used to simplify Boolean algebra expressions.
14. **What is a minterm?**

    * A product term in a Boolean expression where each variable appears exactly once, either in its true or complemented form.
15. **What is a maxterm?**

    * A sum term in a Boolean expression where each variable appears exactly once, either in its true or complemented form.
16. **What is SOP (Sum of Products)?**

    * A form of Boolean expression where several product terms are summed (ORed) together.
17. **What is POS (Product of Sums)?**

    * A form of Boolean expression where several sum terms are multiplied (ANDed) together.
18. **What is a combinational circuit?**

    * A digital circuit whose output depends only on the current inputs.
19. **Give examples of combinational circuits.**

* Adders, Subtractors, Multiplexers, Demultiplexers, Encoders, Decoders.

20. **What is a sequential circuit?**

    * A digital circuit whose output depends on both current inputs and past inputs (state).

21. **Give examples of sequential circuits.**

    * Flip-flops, Latches, Registers, Counters.

22. **What is a flip-flop?**

    * A bistable multivibrator, a circuit that has two stable states and can be used to store state information. It is the basic storage element in sequential logic.

23. **What are the types of flip-flops?**

    * SR, D, JK, T.

24. **Difference between latch and flip-flop?**

    * Latches are level-triggered (sensitive to input level), flip-flops are edge-triggered (sensitive to input transition).

25. **What is a register?**

    * A group of flip-flops used to store multiple bits of data.

26. **What is a counter?**

    * A sequential circuit that counts the number of clock pulses.

27. **What is a multiplexer (MUX)?**

    * A combinational circuit that selects one of many input signals and forwards it to a single output line.

28. **What is a demultiplexer (DEMUX)?**

    * A combinational circuit that takes a single input and routes it to one of many possible output lines.

29. **What is an encoder?**

    * A combinational circuit that converts an active input signal into a coded output signal.

30. **What is a decoder?**

    * A combinational circuit that converts a coded input signal into a set of unique output signals.

31. **What is an adder?**

    * A digital circuit that performs addition of numbers.

32. **What is a half adder?**

* A combinational circuit that adds two single binary digits and produces a sum and a carry.

33. **What is a full adder?**

    * A combinational circuit that adds three single binary digits (two inputs and a carry-in) and produces a sum and a carry-out.

34. **What is a ripple-carry adder?**

    * An adder where the carry-out of each full adder is connected to the carry-in of the next full adder.

35. **What is a carry-lookahead adder?**

    * An adder that improves speed by reducing the time required to determine carry bits.

36. **What is a comparator?**

    * A combinational circuit that compares two binary numbers and determines if they are equal, or if one is greater than the other.

37. **What is a PLA (Programmable Logic Array)?**

    * A type of programmable logic device used to implement combinational logic circuits.

38. **What is a PAL (Programmable Array Logic)?**

    * A type of programmable logic device with a programmable AND array and a fixed OR array.

39. **What is a FPGA (Field-Programmable Gate Array)?**

    * An integrated circuit designed to be configured by a customer or a designer after manufacturing.

40. **What is a CPLD (Complex Programmable Logic Device)?**

    * A programmable logic device with complexity between PALs and FPGAs.

41. **What is HDL (Hardware Description Language)?**

    * A specialized computer language used to describe the structure and behavior of electronic circuits. Examples: VHDL, Verilog.

42. **What is a truth table?**

    * A mathematical table used in logic to compute the functional values of logical expressions on each of their functional arguments.

43. **What is a state diagram?**

    * A diagram that describes the behavior of a state machine, showing states, transitions, and actions.

44. **What is a state table?**

* A table that lists all possible states of a sequential circuit, along with the next state and output for each input combination.

45. **What is synchronous sequential circuit?**

    * A sequential circuit where the state changes only at discrete instants of time, synchronized by a clock signal.

46. **What is asynchronous sequential circuit?**

    * A sequential circuit where the state can change at any time, not synchronized by a clock signal.

47. **What is setup time?**

    * The minimum amount of time that the data input to a flip-flop must be stable before the clock edge.

48. **What is hold time?**

    * The minimum amount of time that the data input to a flip-flop must remain stable after the clock edge.

49. **What is propagation delay?**

    * The time required for a signal to travel from the input to the output of a logic gate or circuit.

50. **What is fan-in and fan-out?**

    * **Fan-in:** The number of inputs a logic gate can accept. **Fan-out:** The number of gates that can be driven by the output of a single gate.

51. **What is tri-state logic?**

    * A type of logic that allows an output to be in one of three states: HIGH, LOW, or high-impedance (disconnected).

52. **What is a bus?**

    * A collection of wires through which data is transmitted from one part of a computer to another.

53. **What is a clock signal?**

    * A periodic signal that synchronizes the operations of a digital circuit.

54. **What is a glitch in digital circuits?**

    * A momentary, unwanted pulse in a digital signal.

55. **What is race condition in DLD?**

    * A situation where the output of a logic circuit depends on the order in which inputs change, which can lead to unpredictable behavior.

56. **What is a hazard in combinational circuits?**

    * A momentary fluctuation in the output of a combinational circuit when input variables change, which can cause incorrect operation in sequential circuits.

57. **What is a static hazard?**

    * A hazard where the output momentarily changes when it should remain constant.

58. **What is a dynamic hazard?**

    * A hazard where the output changes more than once when it should change only once.

59. **How to eliminate hazards?**

    * By adding redundant prime implicants to the K-Map.

60. **What is a Gray code?**

    * A binary numeral system where two successive values differ in only one bit.

61. **What is a BCD (Binary Coded Decimal)?**

    * A system of writing numerals that assigns a four-bit binary code to each decimal digit.

62. **What is a parity bit?**

    * An extra bit added to a binary number to ensure that the total number of 1s in the number is either even or odd.

63. **What is Hamming code?**

    * An error-correcting code that can detect and correct single-bit errors.

64. **What is a finite state machine (FSM)?**

    * A mathematical model of computation used to design sequential logic circuits. It consists of a finite number of states, transitions between states, and actions.

65. **Difference between Mealy and Moore machines?**

    * **Mealy:** Output depends on current state and current input. **Moore:** Output depends only on the current state.

66. **What is a shift register?**

    * A cascade of flip-flops, sharing the same clock, in which the output of each flip-flop is connected to the data input of the next flip-flop in the sequence.

67. **What is a memory cell?**

    * The fundamental building block of computer memory, capable of storing a single bit of binary information.

68. **What is ROM (Read-Only Memory)?**

    * Non-volatile memory that stores data permanently and cannot be easily modified.

69. **What is RAM (Random Access Memory)?**

    * Volatile memory that stores data temporarily and can be read from and written to.

70. **Difference between SRAM and DRAM?**

    * **SRAM (Static RAM):** Faster, more expensive, uses latches, doesn't need refreshing. **DRAM (Dynamic RAM):** Slower, cheaper, uses capacitors, needs refreshing.

System Architecture

1. **What is System Architecture?**
    - The conceptual model that defines the structure, behavior, and more views of a system. It describes how the components interact and function together.
2. **What are the main components of a computer system?**
    - CPU, Memory (RAM, ROM), Storage (HDD, SSD), Input/Output devices, Bus.
3. **What is the CPU (Central Processing Unit)?**
    - The electronic circuitry within a computer that carries out the instructions of a computer program by performing the basic arithmetic, logic, controlling, and input/output (I/O) operations.
4. **What are the main components of a CPU?**
    - ALU (Arithmetic Logic Unit), Control Unit, Registers.
5. **Explain ALU.**
    - Performs arithmetic operations (addition, subtraction, etc.) and logical operations (AND, OR, NOT).
6. **Explain Control Unit.**
    - Manages and coordinates the components of the computer system. It fetches, decodes, executes, and stores instructions.
7. **What are registers?**
    - Small, high-speed storage locations within the CPU that hold data temporarily during processing.
8. **What is a program counter (PC)?**
    - A CPU register that holds the address of the next instruction to be executed.
9. **What is an instruction register (IR)?**
    - A CPU register that holds the instruction currently being executed.
10. **What is a memory address register (MAR)?**

* A CPU register that stores the memory address from which data will be fetched to the CPU, or the address to which data will be sent and stored.

11. **What is a memory data register (MDR)?**

    * A CPU register that stores data that is transferred to and from the immediate access storage.

12. **What is the Von Neumann architecture?**

    * A computer architecture where program instructions and data are stored in the same memory space.

13. **What is the Harvard architecture?**

    * A computer architecture with physically separate storage and signal pathways for instructions and data.

14. **Difference between Von Neumann and Harvard architecture?**

    * Von Neumann uses a single memory for both instructions and data, Harvard uses separate memories. Harvard allows simultaneous fetching of instructions and data.

15. **What is a bus? Types of buses?**

    * A communication system that transfers data between components inside a computer. Types: Data bus, Address bus, Control bus.

16. **What is cache memory?**

    * A small, fast memory that stores copies of data from frequently used main memory locations to reduce average data access time.

17. **Explain different levels of cache (L1, L2, L3).**

    * **L1 Cache:** Smallest, fastest, closest to CPU core. **L2 Cache:** Larger, slower than L1, shared by cores. **L3 Cache:** Largest, slowest, shared by all cores on the CPU.

18. **What is cache hit and cache miss?**

    * **Cache Hit:** When requested data is found in the cache. **Cache Miss:** When requested data is not found in the cache and must be retrieved from main memory.

19. **What is cache coherence?**

    * Ensuring that all copies of shared data in multiple caches are consistent.

20. **What is virtual memory?**

    * A memory management technique that allows a computer to compensate for physical memory shortages by temporarily transferring data from RAM to disk storage.

21. **What is paging?**

    * A memory management scheme that allows the physical address space of a process to be non-contiguous. It divides physical memory into fixed-size blocks called frames and

logical memory into blocks of the same size called pages.

22. **What is a page table?**

    * A data structure used by a virtual memory system in a computer operating system to store the mapping between virtual addresses and physical addresses.

23. **What is a TLB (Translation Lookaside Buffer)?**

    * A CPU cache that stores recent translations of virtual memory to physical addresses, used to speed up virtual address translation.

24. **What is an interrupt?**

    * A signal to the processor emitted by hardware or software indicating an event that needs immediate attention.

25. **What is DMA (Direct Memory Access)?**

    * A feature of computer systems that allows certain hardware subsystems to access main system memory (RAM) independently of the central processing unit (CPU).

26. **What is pipelining in CPU?**

    * A technique that allows the CPU to execute multiple instructions concurrently by overlapping the execution phases of different instructions.

27. **What is instruction-level parallelism (ILP)?**

    * The ability to execute multiple instructions in parallel within a single processor.

28. **What is superscalar architecture?**

    * A CPU architecture that implements instruction-level parallelism within a single processor, allowing it to execute more than one instruction per clock cycle.

29. **What is a multi-core processor?**

    * A single computing component with two or more independent actual processing units (cores).

30. **What is multithreading in CPU architecture?**

    * The ability of a CPU to execute multiple threads (parts of a program) concurrently within a single core.

31. **Difference between multi-core and multithreading?**

    * Multi-core involves multiple physical processing units. Multithreading involves multiple logical threads of execution within a single core.

32. **What is a GPU (Graphics Processing Unit)?**

    * A specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device.

33. **Difference between CPU and GPU?**

    * CPU is optimized for sequential processing and complex tasks. GPU is optimized for parallel processing and simple, repetitive tasks.

34. **What is an ASIC (Application-Specific Integrated Circuit)?**

    * An integrated circuit customized for a particular use, rather than intended for general-purpose use.

35. **What is an FPGA (Field-Programmable Gate Array)?**

    * An integrated circuit designed to be configured by a customer or a designer after manufacturing.

36. **Difference between ASIC and FPGA?**

    * ASICs are custom-designed for a specific function, offering high performance and efficiency but high NRE costs. FPGAs are reconfigurable, offering flexibility but lower performance/efficiency.

37. **What is RISC (Reduced Instruction Set Computer)?**

    * A CPU design philosophy that favors a smaller, simpler, and highly optimized set of instructions.

38. **What is CISC (Complex Instruction Set Computer)?**

    * A CPU design philosophy that favors a larger, more complex set of instructions, often performing multiple operations in a single instruction.

39. **Difference between RISC and CISC?**

    * RISC has fewer, simpler instructions, executed faster. CISC has more, complex instructions, executed slower but fewer instructions per task.

40. **What is an embedded system?**

    * A computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.

41. **What is an IoT (Internet of Things) device?**

    * A physical object that is embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.

42. **What is a distributed system?**

    * A system whose components are located on different networked computers, which communicate and coordinate their actions by passing messages to one another.

43. **What is a client-server architecture?**

* A distributed application architecture that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

44. **What is a peer-to-peer architecture?**

    * A decentralized communications model in which each party has equivalent capabilities and can initiate a communication session.

45. **What is a cloud computing?**

    * The on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user.

46. **What are the service models of cloud computing?**

    * IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service).

47. **What are the deployment models of cloud computing?**

    * Public cloud, Private cloud, Hybrid cloud, Community cloud.

48. **What is a microkernel architecture?**

    * An operating system architecture where only the most essential services (e.g., process management, memory management) run in kernel space, while other services run in user space.

49. **What is a monolithic kernel architecture?**

    * An operating system architecture where all operating system services (e.g., process management, memory management, file system, device drivers) run in kernel space.

50. **What is a hypervisor?**

    * A software, firmware, or hardware that creates and runs virtual machines (VMs). Also known as a Virtual Machine Monitor (VMM).

51. **What is virtualization?**

    * The creation of a virtual (rather than actual) version of something, such as a server, a storage device, a network resource, or even an operating system.

52. **What is containerization?**

    * A form of operating system virtualization in which an application or service is packaged into a container image, which can then be deployed and run in an isolated environment.

53. **Difference between virtualization and containerization?**

    * Virtualization virtualizes the hardware, running multiple OS instances. Containerization virtualizes the OS, sharing the host OS kernel.

54. **What is a network topology?**

* The arrangement of the elements (links, nodes, etc.) of a communication network.

55. **What are common network topologies?**

    * Bus, Star, Ring, Mesh, Tree, Hybrid.

56. **What is the OSI model?**

    * Open Systems Interconnection model. A conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system without regard to their underlying internal structure and technology.

57. **List the layers of the OSI model.**

    * Physical, Data Link, Network, Transport, Session, Presentation, Application.

58. **What is the TCP/IP model?**

    * A conceptual model and a set of communications protocols used in the Internet and similar computer networks. It is a simpler model than OSI.

59. **List the layers of the TCP/IP model.**

    * Network Access, Internet, Transport, Application.

60. **What is a router?**

    * A networking device that forwards data packets between computer networks. Routers perform the traffic directing functions on the Internet.

61. **What is a switch?**

    * A networking device that connects devices in a computer network and uses packet switching to receive, process, and forward data to the destination device.

62. **Difference between router and switch?**

    * Routers connect different networks, switches connect devices within a single network.

63. **What is a gateway?**

    * A network node used in telecommunications that connects two networks with different transmission protocols together.

64. **What is a firewall?**

    * A network security device that monitors and filters incoming and outgoing network traffic based on an organization's previously established security policies.

65. **What is a load balancer?**

    * A device that distributes network traffic evenly across multiple servers to ensure high availability and reliability.

66. **What is a proxy server?**

* A server that acts as an intermediary for requests from clients seeking resources from other servers.

67. **What is a CDN (Content Delivery Network)?**

    * A geographically distributed network of proxy servers and their data centers. The goal is to provide high availability and performance by distributing the service spatially relative to end-users.

68. **What is a database management system (DBMS)?**

    * A software system designed to manage and facilitate the organization, storage, retrieval, security, and integrity of data.

69. **What is a relational database?**

    * A type of database that stores and provides access to data points that are related to one another. Data is organized into tables.

70. **What is NoSQL database?**

    * A non-relational database that provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

71. **What is ACID properties in databases?**

    * **Atomicity:** All or nothing. **Consistency:** Valid state before and after transaction. **Isolation:** Concurrent transactions appear to execute serially. **Durability:** Committed transactions persist.

72. **What is BASE properties in NoSQL databases?**

    * **Basically Available:** System guarantees availability. **Soft state:** State may change over time even without input. **Eventually consistent:** System will eventually become consistent.

73. **What is a data warehouse?**

    * A large collection of business data used to help an organization make decisions. It is a central repository of integrated data from one or more disparate sources.

74. **What is a data lake?**

    * A centralized repository that allows you to store all your structured and unstructured data at any scale.

75. **What is an API Gateway?**

    * A server that acts as an API front-end, takes a single entry point for all clients, and routes requests to the appropriate microservice or backend service.

76. **What is a Message Broker?**

* An intermediary computer program module that translates a message from the formal messaging protocol of the sender to the formal messaging protocol of the receiver.

77. **What is a Service Mesh?**

    * A dedicated infrastructure layer for handling service-to-service communication. It ensures that communication among services is fast, reliable, and secure.

78. **What is Serverless Architecture?**

    * A cloud-native development model that allows developers to build and run applications without having to manage servers.

79. **What is Edge Computing?**

    * A distributed computing paradigm that brings computation and data storage closer to the sources of data.

80. **What is Quantum Computing?**

    * A new type of computing that harnesses the phenomena of quantum mechanics, such as superposition and entanglement, to perform computations.

81. **What is a Quantum Bit (Qubit)?**

    * The basic unit of quantum information, analogous to the binary bit in classical computing.

82. **What is a Quantum Gate?**

    * A basic quantum circuit operating on a small number of qubits.

83. **What is a Quantum Algorithm?**

    * An algorithm that runs on a realistic model of quantum computation, such as the quantum circuit model.

84. **What is Shor's Algorithm?**

    * A quantum algorithm for integer factorization, which can break many public-key cryptosystems.

85. **What is Grover's Algorithm?**

    * A quantum algorithm that speeds up the search for a target item in an unstructured database.

86. **What is a Quantum Supremacy?**

    * The point where quantum computers can perform tasks that are practically impossible for classical computers.

87. **What is a Quantum Annealing?**

    * A quantum computing technique used for finding the global minimum of a given objective function over a set of candidate solutions.

88. **What is a Quantum Key Distribution (QKD)?**

    * A secure communication method that implements a cryptographic protocol involving components of quantum mechanics.

89. **What is a Quantum Internet?**

    * A network that would allow quantum devices to exchange information using quantum mechanics.