Interview doc

Java Questions

Q) Explain the “this” keyword, under what assumptions does the this keyword not work?
- Refers to the current instance of a class.
- Does not work with static methods/fields.

Q) Explain the static keyword, what’s it for, why is it in the main function
- Static variable: Shared among all instances.
- Static method: Belongs to the class, callable without an instance.
- Static block: Executes when the class is loaded.
- Main function: Static to allow JVM to call it without creating an instance.

Q) Explain public and private what’s the concept behind them
- Access specifiers.
- Public: Accessible from anywhere.
- Private: Accessible only within the declared class.

Q) How is an output statement like system.out.println() in Java better than a statement like print in python?
- Not necessarily better, just different.
- Java provides detailed I/O libraries; output method choice depends on context and needs.

Q) What is the difference between a class and instance?
- Class: Blueprint for objects.
- Instance: Concrete object created from the class.

Q) What is an interface, can we implement interfaces in Java?
- Interface: Abstract type used to specify a behavior.
- Yes, we can implement interfaces.

Q) Why does the main function have to be public? Why is it public static void main?
- Public: JVM needs to access it from anywhere.
- Static: Invoked without creating an instance.
- Void: Does not return any value.
- Main: Entry point for the program.

Q) Why do we need a main function in the first place as a mandatory thing in Java? Suppose we just installed Java and we want to write a hello world program, to just do that we need the class

and main fn, why?
- Main function: Entry point for execution.
- Class structure enforces encapsulation and organization.

Q) Name other object-oriented languages except Java
- C++, Python, C#

Q) What is diamond inheritance?
- Hybrid inheritance where we see Hierarchical inheritance (first) and Multiple inheritance (second). In the top half, two classes inherit from the same parent, and in the bottom half, one class inherits from both the parents.

Q) Give an example of polymorphism
- Method overloading or method overriding.

Q) What is method overloading and how is it different from method overriding. Give the syntax for both
- Overloading: Same method name, different parameters.

```
void display(int a) { }
void display(int a, int b) { }
```

- Overriding: Subclass provides specific implementation of a method declared in the parent class.

```
@Override
void display() { }
```

Q) How to implement multi threading in Java? Give syntax. What is thread synchronization? What are the various thread synchronization techniques?
- Extend `Thread` class or implement `Runnable` interface.

```
  class MyThread extends Thread {
    public void run() { }
  }
```

- Synchronization: Ensures only one thread accesses a resource at a time.

```
  synchronized (this) { }
```

- Techniques: Locks, synchronized blocks/methods, `java.util.concurrent` locks.

Q) What are constructors? How do they work?
- Special methods to initialize objects.

```
  MyClass() { }
```

Q) Explain the SOLID principles in OOPS
- S: Single Responsibility Principle
- O: Open/Closed Principle
- L: Liskov Substitution Principle
- I: Interface Segregation Principle
- D: Dependency Inversion Principle

Q) What are UML diagrams? Explain some
- Visual representation of a system's architecture.
- Class diagrams, sequence diagrams, use case diagrams.

Q) How is exception handling done in Java? Explain try, catch, finally
- Try: Code that may throw an exception.
- Catch: Handle the exception.
- Finally: Executes after try/catch, regardless of an exception.

```
try { }
catch (Exception e) { }
finally { }
```

Q) What is garbage collection in Java and how does it work?
- Automatic memory management.
- JVM reclaims memory used by unreachable objects.

Q) Can you explain what is the JVM, JRE and JDK. And give a brief description of the diagram.
- JVM: Runs Java bytecode.
- JRE: Environment to run Java applications.
- JDK: Tools for developing Java applications.

Q) Explain some features of Java
- Platform-independent, Object-oriented, Robust, Secure, Multithreaded.

Q) What are the different data types supported in Java? Explain how many bits or bytes each one occupies
- Primitive types: `byte` (1 byte), `short` (2 bytes), `int` (4 bytes), `long` (8 bytes), `float` (4 bytes), `double` (8 bytes), `char` (2 bytes), `boolean` (1 bit).

Q) What is an abstract class? Give me an example. What is the difference between an abstract class and an interface?
- Abstract class: Cannot be instantiated, can have abstract and concrete methods.

```
abstract class MyClass
```

```
{
   abstract void myMethod();
}
```
- Interface: Only abstract methods (Java 8 onwards, can have default methods).

Q) What is an array list in Java? Can you explain the difference between array list and linked list?
- ArrayList: Resizable array.
- LinkedList: Doubly-linked list.
- Difference: ArrayList allows random access, LinkedList is better for frequent insertions/deletions.

Q) What are some commonly used Java libraries? Java frameworks, packages?
- JUnit, Spring, util, lang

Q) What is string builder and string buffer classes? What is an enum class?
- StringBuilder: Mutable sequence of characters, not thread-safe.
- StringBuffer: Mutable sequence of characters, thread-safe.
- Enum: Special class representing a group of constants.

Q) What are streams in Java?
- Stream API: Process sequences of elements.

Q) What is Java spring? What is Java spring boot? Differences?
- Spring: Framework for enterprise Java applications.
- Spring Boot: Simplifies Spring application setup with auto-configuration.

Q) What is dependency injection and how is it used in Spring?
- DI: Design pattern to inject dependencies.
- Spring uses annotations like `@Autowired` to manage dependencies.

Q) What are the common frameworks used for testing in Java?
- JUnit, Mockito.

Q) What is maven? What is gradle?
- Maven: Build automation tool using XML.
- Gradle: Build automation tool using Groovy/Kotlin DSL.

Q) What are constructors? How do they work? Can you implement more than one constructor in a class?
- Special methods to initialize objects.
- Yes, through constructor overloading.

```
  MyClass() { }
  MyClass(int a) { }
```

Database Questions

Q) Explain CRUD operations and examples of each
- Create: `INSERT INTO table_name VALUES (...)`
- Read: `SELECT * FROM table_name`
- Update: `UPDATE table_name SET column=value`
- Delete: `DELETE FROM table_name WHERE condition`

Q) Explain the ACID properties in detail
- Atomicity: All-or-nothing transactions.
- Consistency: Transactions leave the database in a valid state.
- Isolation: Transactions do not interfere with each other.
- Durability: Committed transactions are permanent.

Q) What is an SQL injection?
- Code injection attack to manipulate or exploit databases.

Q) Explain the difference between SQL and NoSQL
- SQL: Relational databases, structured schema.
- NoSQL: Non-relational, flexible schema, handles unstructured data.

Q) List the different keys in databases. What is a foreign key
- Primary Key: Unique identifier.
- Foreign Key: References primary key in another table.

Q) What is normalization? What are normal forms? Why do we need normalization?
- Process of organizing data to reduce redundancy.
- Normal forms: 1NF, 2NF, 3NF, BCNF.
- Ensures data integrity and efficiency.

Q) What is denormalization
- Process of combining tables to improve read performance.

Q) Explain the conditions for 3.5NF
- A table is in BCNF if every determinant is a candidate key.

OS Questions

Q) What is a deadlock? What are some deadlock resolution methods? What is live lock? What is the difference between deadlock and circular dependency? Explain about deadlock detection, avoidance, prevention, recovery mechanisms/algorithm.
- Deadlock: Two or more processes unable to proceed.
- Resolution: Deadlock prevention, avoidance (Banker's algorithm), detection, and recovery.
- Livelock: Processes keep changing states without progress.
- Circular dependency is a necessary condition for deadlock.

Q) What is a resource allocation graph? What is a Gantt chart?
- Resource Allocation Graph: Represents resources allocated to processes.
- Gantt Chart: Visual representation of a schedule.

Q) What is the difference between a process and a thread?
- Process: Independent program in execution.
- Thread: Smallest unit of a process, shares memory.

Q) Explain various process states
- New, Ready, Running, Waiting, Terminated.

Q) What is context switching
- Switching CPU from one process/thread to another.

Q) What is virtual memory and how does it work?
- Virtual memory is a memory management technique used by operating systems.
- It allows a computer to temporarily increase the capacity of its main memory by using secondary memory as though it were part of the main memory.
- Extends physical memory using disk storage.

Q) What are paging and segmentation?
- Paging: memory management technique by which a computer stores and retrieves data from secondary storage for use in main memory. Memory divided into fixed-size pages.
- Segmentation: Memory management technique that divides Memory divided into variable-sized segments.

Q) What are the differences between physical and virtual addresses?
- Physical Address: Actual location in memory.
- Virtual Address: Address used by the process.

Q) What is a critical section
- Code segment where shared resources are accessed. To ensure deadlock avoidance no 2 processes must execute within the critical section at the same time.

Q) What are system calls? Give me an example
- System call is the programmatic way in which a computer program requests a service from the operating system on which it is executed.
- Eg: fork() - to create a new process, exec() - to execute a new program in the current process, exit() - to terminate current process

Q) What is Interprocess communication? State some types
- Mechanism for processes to communicate among each other.
- Types: Message passing, Shared memory, sockets.

Networks Questions

Q) Can you explain the OSI Model and its layers?
- Layers: Physical, Data Link, Network, Transport, Session, Presentation, Application.

Q) What is the difference between TCP and UDP?
- TCP: Connection-oriented, reliable.
- UDP: Connectionless, faster but less reliable.

Q) What is a subnet and how does subnetting work?
- Subnet: Subdivision of an IP network.
- Subnetting: Divides a network into smaller subnets.

Q) How do switches differ from routers?
- Switch: Connects devices in a network.
- Router: Connects different networks.

Q) Can you explain what a firewall is and how it works?
- Security device that monitors and controls network traffic.

Q) What is DNS?
- Domain Name System: Translates domain names to IP addresses.

Q) What is the difference between IPv4 and IPv6?
- IPv4: 32-bit addresses.
- IPv6: 128-bit addresses, more address space.

Q) What is a VPN

- Virtual Private Network: Secures connection over the internet by encrypting data.