Software Engineering Basics Competency

Version Control / Git

- What is Git?
  - A revision control system.
  - git – the stupid content tracker
- Why do we use Git?
  - Allows for incremental modification.
  - Allows for testing and collaborating on issues: branch/merge.
- What is a commit?
  - A commit records changes staged (using git add).
- What is GitHub?
  - Most popular public git server.
- What does pushing do?
  - Updates remote repository using local commits.
- What does pulling do?
  - Adds changes to local repositor from remote.
- What does clone do?
  - Clones a remote repositor to a local one.
- What does branch do?
  - Creates a new branch.
- What does checkout do?
  - Switches to a specific branch.
- What does merge do?
  - Adds changes from one branch to the current branch.
- What is a merge conflict?
  - When the two branches try to modify the same content, it's unclear which branch's modification should be used.

Agile Development

- What is the SDLC?
  - Software Development Lifecyle is the process that development teams use to plan the development of an application to make teams more efficient and deliver quality products.
- What is Agile development?
  - It's a form of SDLC that is used for rapid development of a product with daily standups to ensure the quality standards are being met as well as sprints to develop software rapidly with the option to improve things as you go.
- What is a Sprint?

  - A sprint is a short amount of time in which the team works to complete a set amount of work.
- What are Ceremonies in agile/scrum?
  - Sprint planning, daily standups, sprint meetings(review), retrospective
- What are user stories?
  - Product owner requirements, what the client wants to see in their product.
- What is story pointing?
  - An estimation from the development team on how much work or how difficult a task may be, stories are given points based on their difficulty.
- What is velocity in agile development?
  - How many user stories the team managed to complete during a sprint

UNIX

- What is UNIX?
  - Unix is an operating system developed originally for the PDP-11 computer that can do multiuser, multitasking.
  - The basis for Linux, BSD, MacOS and iOS.
- How do I change directories in UNIX?
  - To change directories on UNIX with “cd” command
- How do I view the contents of my directory in UNIX?
  - To view directory contents on UNIX with “ls“ command. Here are a few optional arguments for ‘ls’.

    -a to see hidden files.

    -l to see more information about the file(s).

    -lh human readable information.

    -r to list files in reverse order.

    -t sort by time, newest first.

Java Competency

Java basics

- What is compilation?
  - The translation of user written .java files to JVM executable byte-code (.class files).
- What does it mean for Java to be strongly typed?
  - When a variable is declared in Java, a type must be specified
- What are primitive types?
  - Primitive types are data types which all other data types are constructed

- What are the 8 primitive types in Java?
  - Short, Long, Double, Byte, Char, Int, Float & Boolean
- What is a method?
  - A block of code that runs when it's called.
- What does 'return' do?
  - Returns the value of a method
- What is a return type?
  - A data type of the value that is returned from the method.
- What does the return type 'void' mean?
  - Return type void means the method should not have a return value
- What is a method parameter?
  - Are a list of variables in a method declaration
- What are the different boolean operators?
  - And, Or & Not
- What are Strings in Java?
  - Arrays of characters for storing printable text.
- What is a stack trace?
  - A collection of records that stores movement during execution, often used for debugging and tells you where in a code an exception has occurred.
- What is the main method? What is its syntax?
  - The main method is the entry point to the program and is where the JVM begins execution.

Java architecture

- What is the JDK?
  - Java Development Kit. A software development kit that develops applications in Java.
- What is the JRE?
  - Java Runtime Environment. A software package that contains class libraries of Java.
- What is the JVM?
  - Java Virtual Machine. Responsible for providing all implementations to the JRE.
  - Allows java byte code to run on any system with the JVM.
- What terminal command is used to compile a Java file?
  - javac
- What is contained in stack memory?
  - Used for execution of a thread. Contains stack frames (local variables, a return address and method parameters); method specific values and references to objects in heap space. Referenced in LIFO (Last-In-First-Out) order.
- What is contained in heap memory?

    - Objects and JRE classes. Whenever an object is created, it is created in heap space.
    - Dynamically allocated data.
- What is the String Pool & does it belong to stack or heap memory?
    - An immutable pool of Strings that belong in heap memory.
- What is garbage collection?
    - Automated process of deleting dynamically allocated memory that’s no longer referenced (no dangling pointers, memory leaks or double frees).

OOP basics

- What is OOP?
    - Object oriented programming is the style of programming focusing on instantiating classes as objects and modifying multiple aspects separately
- What are Objects?
    - An object is the instantiated class that contains multiple variables and is mutable through that class's methods
- What makes an Object different from a primitive type?
    - Primitive types are pre-defined where objects are user-defined
- What is the relationship between a Class and an Object in Java?
    - A class is a template to create an object
- What are constructors?
    - A constructor is a method with the same name as the class it is in that is used to initialize an object’s variables when called
- What is the default constructor?
    - The default constructor is a constructor automatically used to assign default values to an object

Control flow

- What is an Array?
    - An array is a collection or elements of the same type with a fixed length
- How do I get an element of an array?
    - An element can be retrieved with the format ‘arrayName[index]’ with index starting at 0 and going to the array length-1
- What are the different flow control statements in Java?
    - If-then, if-then-else, switch, while, do-while, for, break, return, and continue
- How is a for loop written in Java?
    - for(variable; condition; increment/decrement){//code}
- What is the difference between ++i and i++?
    - ++j increments then returns where j++ increments after.
- What is the difference between while and do-while loops?

    - A while loop has a condition that must be fulfilled before the loop logic code will be run, a do-while loop always executes the first loop and then checks the condition to keep looping
- What are break & continue statements?
    - Break is used to exit a loop completely, continue is used to exit the current iteration and proceed to the next

Exceptions

- What are exceptions in Java?
    - a general exception class which provides an abstraction for all exceptions
- How are errors different from exceptions?
    - Exceptions are problems in the code that a program should try to catch where an error is a serious problem that cannot be catched and are fatal for the program and out of the programmer's control
- What is the difference between checked and unchecked exceptions?
    - Checked exceptions are checked at compilation where unchecked are checked at runtime
- What might cause a NullPointerException?
    - When a variable being accessed has no value being pointing to null
- Is ArrayIndexOutOfBoundsException a runtime exception?
    - It is an unchecked runtime exception
- Is FileNotFoundException a runtime exception?
    - It is a checked compilation exception, not runtime
- How do I find where an exception was thrown within the program?
    - Using the exception stack trace
- What does 'throws' do?
    - Throws defines the type of exceptions a method can throw
- What does try/catch do?
    - Try/catch is used around code that may generate exceptions
- Can I have multiple catch blocks? Multiple try blocks?
    - There can be multiple catch blocks per try ordered from the most specific to general exceptions the code can generate, there can be nested try blocks as well but each of them must have their own catch blocks

Collections

- What are collections in Java?
    - Framework that provides an architecture to store and manipulate groups of objects.
- What is the difference between a List and a Set?
    - A List is an indexed sequence, a Set is a non-indexed, non-duplicate sequence.
- What is the difference between a Set and a Map?

  - A Set is unordered and contains unique elements, a Map contains data with key-value pairs.
- What is the difference between a Stack and a Queue?
  - A Stack follows LIFO (Last In First Out) order while A Queue follows FIFO (First In First Out) order.
- What is the difference between an ArrayList and LinkedList?
  - An ArrayList is a dynamic array that stores elements, a LinkedList internally uses double linked list to store elements.
- Are maps part of the collection interface?
  - No, but a Map is a part of the Collections Framework, not the Interface.
- What is a wrapper class?
  - Provides a way to use primitive data types as objects. (Ex. int -> Integer)

Java keywords & features

- What do access modifiers do?
  - Keywords in Java that provide accessibility of a class and it's member. They set the access level to methods, variable, classes and constructors.
- What are the 4 access modifiers?
  - Public, default, protected and private.
- What are the non-access modifiers in Java?
  - Static, Final and Abstract.
- What does Static do?
  - Static keyword is used to create variables or methods that will exist independently of any instance created for the class, can be accessed from any other class.
- What does final do?
  - For finalizing the implementations of classes, methods and variables.
- What is Scope in programming languages?
  - The area where a method or variable is accessible in a program.
- What are the different scopes in Java?
  - Method level, Class level, Block scope. global/static
-

Maven

- What is Maven?
  - Maven is a project management and comprehension tool that provides developers a complete build lifecycle framework.
- What file should be changed to add new Maven dependencies?
  - pom.xml
- What is the Maven lifecycle?
  - When Maven builds your project, it goes through several steps called phases, which are the following in order:

        - Validate, Compile, Test, Package, Integration, Verify, Install, Deploy.
- How do I find & add a new dependency to Maven?
    - We can search for new maven dependencies on central.sonatype.com, then in the dependencies section of our pom.xml file, simply paste the dependency details found.

Testing

- What is JUnit?
    - JUnit is an open-source framework used for writing and executing test in Java.
- What is a unit test?
    - Unit Test is used to verify code is working as expected.
- What are some of the annotations used in JUnit?
    - @test – Tells JUnit the public void method that its attached to can be run as a test case.
    - @BeforeAll – Tells JUnit to run a method once before all tests
    - @BeforeEach – tells JUnit to run a method before each test
    - @AfterAll – Tells JUnit to run a method after all tests have completed
    - @AfterEach – Tells JUnit to run a method after each test
- What is TDD?
    - Test-Driven Development – consists of writing unit test first before the code has been written.
- What is Mockito for?
    - Mockito is used to simplify the development of test for classes with external dependencies
- How are Mock objects in Mockito created?
    - By using the mock() method which creates a mock object of a given class.

Algorithms

- What is time complexity?
    - An estimate of how long an algorithm will take to execute on different input sizes.
- What makes an algorithm O(1), O(n), O(log(n)), or O(n^2)?
    - O(1) -> Constant time, algorithm will take the same amount of time regardless of input size.
    - O(n) -> Linear time, execution time scales directly with input size.
    - O(log(n)) -> Logarithmic time, each time the size input doubles, execution time increases by the same amount.
    - O(n^2) -> Quadratic time, the algorithm scales by the input size's square.
- Could you describe the linear search algorithm? What is its time complexity?
    - Linear search as name implies is linear time (O(n)).
    - Iterate over every element in the collection until you find a match.

- Could you describe the binary search algorithm? When can you use it & what is its time complexity?
    - Logarithmic time (O(log(n))).
    - Compare the middle element and search term, then:
        - Less than -> Do a binary search on the lower half (excluding middle).
        - Greater than -> Do a binary search on the upper half (excluding middle).
        - Equal -> Return middle.
- What steps could you take to sort an array?
    - Ambiguous question
- How does an ArrayList work?
    - Use dynamically allocated memory to store elements. Also store current number of elements and allocated size.
    - When more elements are added than allocated size you must first:
        - Reallocate memory with some larger size.
        - Update allocated size variable.
- How does a LinkedList work?
    - Self referential structure.
    - A LinkedList node stores (doubly linked):
        - Some type T.
        - A reference to the next node called next.
        - A reference to the previous node called prev.
    - The previous node's prev reference is null.
    - The last node's next reference is also null.

Advanced OOP

- What are the 4 pillars of OOP?
    - The four pillars of OOP are Abstraction, Encapsulation, Polymorphism, and Inheritance.
- Can you explain Inheritance?
    - Inheritance enables an object to inherit properties and behaviors from its parent object. It enables creating classes in a hierarchical structure consisting of subclasses and derived classes.
- Can you explain Polymorphism?
    - Polymorphism means to take many forms. In programming this concept is used when overriding or overloading methods. Method overriding occurs when a method of the same name accepts a different number or type of arguments. Method overriding occurs when a derived class re-implements a method contained within a base class using the same signature or with a covariant return type. Polymorphism also allows a derived class to be used in the same context as a base class.
- Can you explain Encapsulation?

    - Encapsulation is the process of hiding implementation details. It is the bundling of properties and methods that are unique to a class object. Through encapsulation an instance of a class has its own state and a set of methods that operate on that state that is separate from all other instances of that same class.
- Can you explain Abstraction?
    - Abstraction is the process of generalizing and specializing classes. With generalization, properties and methods that are similar between two different classes are factored out into a base class. With Specialization, classes obtain properties and methods that are unique to themselves. Abstraction simplifies complex systems by creating abstract classes or interfaces that define common behavior shared by numerous other classes.
- What is the Object class in Java?
    - A Java Object is a root class that is shared by all other objects in the Java programming language. It defines a common set of behaviors that all other objects have. Some of these common behaviors include the .equals, .hashCode, and .toString methods.
- What methods does the Object class contain?
    - The Object class contains the .equals and .hashCode methods that are useful for testing equality. It contains a .toString method to convert an object into a string representation of itself. It also contains the .clone method for making copies of itself. Additional methods include the .finalize and .getClass methods. The finalize method is called before an object is garbage collected and the .getClass method returns the run time representation of a class using Java's reflection API.
- What are Generics in Java?
    - Generics in Java enable the use of type as a parameter. By using Generics, a programmer can create type agnostic code that works on more than one type of object while also maintaining type safety. Generics can also be bounded to restrict what type of objects can be passed to it.
- What are interfaces in Java?
    - Java interfaces define a set of abstract methods and constants that can be implemented by a base class. They can be thought of as a contract that forces any class that implements an interface to also implement its methods. Interfaces can be used as the type for method parameters and regular variables. Those method parameters and variables can be initialized with a class object that implements that interface.
- What does extending a class do?
    - Extending a class creates a "is-a" relationship. The extended class becomes the base class and the class that is extending becomes the derived, or subclass. Extending a class allows for the recycling of methods and properties from a base class. The derived class may override base class methods or properties to create new behavior.

- What does implementing an interface do?
    - A class that implements an interface is required to implement all of its abstract methods as well. Implementing an interface is a great way to ensure that certain behaviors exist when using object-oriented programming.
- What is the difference between runtime and compile time polymorphism?
    - An example of runtime polymorphism is method overriding. Method overloading is considered compile time polymorphism. With overloading, the correct method to call is decided based on method signature and is decided at compile time. With overriding the method signature must be the exact same but also allows for a covariant return type. A covariant return type is a type that is derived from the type of the original return type.
- What is method overloading?
    - Method overloading is the process of defining different behavior for a method while using the same method name as the original method. At compile type the Java compiler determines which method to call based on the method's parameters.
- What is method overriding?
    - Method overriding, also called runtime polymorphism, is the process of defining new behavior for a method in a derived class when a method of the same name and signature exists in the base class. The derived class must use the same exact method signature as the base class or use a covariant return type.
- Can I extend multiple classes?
    - Java uses single inheritance. This means that only one class at a time can be extended by class in Java.
- Can I implement multiple interfaces?
    - Java supports implementing multiple interfaces at the same time by creating a comma separated list of interfaces named after the implements keyword.
- How might access modifiers help us achieve Encapsulation?
    - Access modifiers such as private and protected are used to prevent foreign objects from accessing the properties or methods of a class. Encapsulation is the process of hiding implementation details. By using access modifiers, a class can keep its implementation hidden.
- How might interfaces help us achieve Abstraction?
    - Interfaces are great for generalizing code that will be used in multiple parts of a program. Abstraction is the process of simplifying classes by factoring out common behavior. Interfaces can define a set of methods that must be implemented in derived classes.
- What does the Comparable interface do?
    - The comparable interface defines an abstract method, .compareTo, that is used to check if another object of the same type is greater, less than, or equal. It is often used when sorting a list of objects. When an object

    - should be sorted before another object it returns negative, if they are the same it returns 0, and it should be sorted after it returns a positive number.

Multithreading

- What is a thread?
    - An independent execution.
- Why would using threading be advantageous?
    - Allows for data parallelism.
    - Allows for asynchronous execution for slow resources.
- How do you create a new thread?
    - Fork a process.
    - Get an ExecutorService instance with Executors.newFixedThreadPool(int).
    - Create a new WaitingTread with new WaitingTread(String, int).
- What is a race condition?
    - When two or more threads attempt to access shared, sensitive data and happen by chance to do so at the same time.
- How would you prevent a race condition?
    - Use an atomic operation that the thread scheduler cannot interrupt.
    - Use synchronized methods which block other synchronized method on the same object.
- What is a deadlock?
    - When multiple processes are bocking each other, none can finish the blocked code and unblock the others.

Java 5/8 features

- What features were added in Java 5?
    - Generics, Autoboxing/Unboxing, Typesafe Enums, Varargs, Static Import, Concurrent Collections, Enhanced For Loop, Copy on Write, Compare and Swap, and Locks.
- What features were added in Java 8?
    - Lambda Expressions, Nashorn, String.join() and Streams
- What is Reflection?
    - Allows an executing Java program to introspect on itself. (Ex. getClass())
- What is a lambda expression?
    - A short block of code that takes in parameters and returns a value.
- What is a functional interface?
    - An interface that only contains a single abstract method. A functional interface can also have default and static methods.
- What are streams?
    - Abstraction of non-mutable collection of functions applied in a order to a set of data. Cannot store elements.

- What are some operations that streams can do?
    - ForEach() which loops over the stream elements,
    - Map() produces a new stream after applying a function to all the stream elements,
    - Collect() provides a way to get something (like a list) out of a stream,
    - Filter() produces a new stream with elements that pass a predicate,
    - FindFirst() returns an Optional for the first entry in the stream,
    - ToArray() returns an array of elements from the stream,
    - FlatMap() Helps "flatten" data structure to simplify further operations,
    - Peek() allows the developer to perform multiple operations on each element in a stream

Design Patterns

- How does the Singleton design pattern work & why would you use it?
    - A Singleton restricts the class to a single instance of the class and must provide a global access point. The Singleton design is memory space "friendly" and the single object can be used repeatedly over the client program.
- How does the Factory design pattern work & why would you use it?
    - A Factory allows the developer to create an object without revealing the underlying code to the client program. The benefits include providing an approach for interface rather than implementation and provides abstraction between implementation and client classes through inheritance.

Logging

- What is logging?
    - A powerful aid in understanding and debugging runtime behavior.
- Why would you use logging?
    - Logs capture and persist runtime data and make it available for analysis.

SQL Competency

SQL Basics

- What is SQL & why is it used?

SQL stands for Structural Query Language used to administer data. It is the most standard language to work on databases.

- What is a table in SQL?

A table in SQL consist of rows and columns. It is used for storing data in the database.

- What are primary keys for?

Primary Keys are constraints that uniquely identify each record in a table.

- What are the sublanguages of SQL?

DDL-> Data Definition Language: Create, Alter, Truncate, Drop, Rename

DML->Data Manipulation Language: Update, Delete, Insert

DCL->Data Control Language: Rollback, Commit, Save point

TCL->Transaction Control Language: Revoke, Grant

DQL->Data Query Language: Select, Filter

- How do I query everything from a table?

  Select * from table;

- How do I insert into a table?

Insert into table_name(column,…)values(values,...);

- How do I update values in a table?

  Update table_name set column_name= value where condition;

- What is the difference between drop, delete, and truncate?

-The drop clause removes all data, including the table from the database server, it is a DDL(Data Definition Language).

-The truncate clause removes all rows from the table but doesn't remove the table. It is a DDL.

-The delete clause removes specified rows from a table. It can contain the where clause, if the Where clause is not specified it deletes all rows. It is DML(Data Manipulation Language)

SQL Querying

- How do I query only the rows that meet some criteria in a table?
  - Using the SELECT statement with the WHERE clause
- How do I sort the results of a query in SQL?
  - To sort the results of a SQL query you can use ORDER BY.
- What do aggregate functions do in SQL?
  - Aggragate functions perform calculations on a set of values and returns a single value.

- What are some of the aggregate functions?
    - Count, Sum, Avg, Min, Max
- What does group by do?
    - Group by is used when rows contain the same values in one or more columns and displays them together
- What does having do?
    - Having is used to further filter the results of a group by query.
- What is an alias in SQL?
    - An alias is a temporary name that is assigned to a column or table to make them easier to address and to make them more readable.

SQL Constraints

- What is a constraint in SQL?
    - A constraint is a restriction enforced on a table. Constraints define the type of data that is inserted or updated from a table.
- What is the not null constraint?
    - A not null constraint ensures that a column does not contain a null value.
- What is the unique constraint?
    - A unique constraint ensures that a column value can only exist once in a table.
- What is a primary key?
    - A primary key combines not null and unique constraints to act as a unique identifier for a column or set of columns.
- What is a foreign key?
    - A foreign key matches its value to a primary key in a different table.
- What is referential integrity?
    - Referential integrity ensures that the relationships between tables are valid. This is typically done through matching foreign keys to primary keys. A foreign key cannot exist that does not match up to another tables primary key.

SQL multiplicity

- What is multiplicity in SQL?
    - The numerical relationship between rows of one table and rows in the other
- What are the different types of multiplicity?
    - One-to-One
    - One-to-Many
    - Many-to-Many
- What do you need to add to have one-to-many multiplicity?
    - A table with a primary key(field) that has a relation with many records on another table.

    - **Example:** You have a customer and order tables. The customer has a primary key field “customer_id”, the order table has a “order_id” and “customer_id” fields. **One** customer can have **many** orders in the order table, meaning many orders might have the same “customer_id”.
- What do you need to add to have many-to-many multiplicity?
    - Two or more fields in a table relate to two or more fields in another table
    - **Example:** You have student and class tables. Student table has “student_id” and “class_id” fields, class table has “id” and “name” fields. In this case, any student from the student table can take any class from the class table.
- How do you modify existing tables?
    - You can use the ALTER TABLE command in SQL
- What is normalization & why do we use it?
    - The process of converting a poorly structured table into two or more well-structured tables
- What characterizes 1st normal form (1nf)?
    - Each table cell should contain a single value AND each record needs to be unique.
- What characterizes 2nd normal form (2nf)?
    - Be in 1NF
    - Single column primary key that does not functionally depend on any subset of candidate key relation
- What characterizes 3rd normal form (3nf)?
    - Be in 2NF
    - There should be no transitive dependency for non-prime attributes. So a transitive dependency is a functional dependency in which X → Z (X determines Z) indirectly, by virtue of X → Y and Y → Z (where it is not the case that Y → X)

SQL Joins

- What is join in SQL?
    - A join is an operation used to combine data tables into a single result
- What is an inner join?
    - An inner join returns all rows matched through both tables.
- What are left/right joins?
    - A left join returns all the rows from the left table and all matching rows from the right. Right join vice versa.
- What is a view in SQL?
    - A view is a retrieval of data in a virtual table. Views are created by assigned by giving a name and specifying the data that you want to be included in the view. A view can be manipulated like a regular table after it is created.
- What are constraints in SQL?

    - A constraint is a rule that you enforce on data that is entered into a table. Some examples of this include:
        - Not Null – column cannot contain null values
        - Unique – each column value can only exist once
        - Primary Key - Unique identifier
        - Foreign Key – matches to primary keys in other tables

JDBC

- What is JDBC?
    - JDBC is a java api that allows you to connect and interact with relational databases.
- What are the different classes/interfaces used in JDBC?
    - Some JDBC classes and interfaces include:
        - Connection – provides methods for creating statements and managing transactions
        - Statement – interface that represents the sql statement that is sent to the database
        - PreparedStatement – extends statement and provides support. Precompiled sql statements that can be executed multiple times with different parameters.
        - ResultSet – interface that represents the results that are returned from a database query.
- What needs to be done to query from the database using JDBC?
    - To query from a database using JDBC you must first establish a connection to the database. Then you can create an sql statement and execute it. With resultset you can pull data from that statement and bring it into your java backend.
- What is a DAO for?
    - The Data Access Object is a design pattern that provides an abstraction layer between the java application and the database. The DAO typically involves crating java classes that correspond to tables in the database. Each class encapsulates the data and provides CRUD operations on that dataset.

PL/SQL

- What is a procedure in PL/SQL?
- What is a trigger in PL/SQL?

TCL

- What are transactions?
    - Some work performed on a database
- What is the advantage of using transactions?
    - Transactions are ACID

- What is ACID (briefly)?
  - ATOMIC: A transaction happens all at once or doesn't happen at all.
  - CONSISTENT: All data will be consistent with predefined constraints.
  - ISOLATED: All transactions are independent from each other.
  - DURABLE: Transaction changes are persistent and will remain even if there is a system crash.

Indexes

- What is an index?
  - It's a table to quickly look up information from other tables that need to be searched up frequently.
- Why is an index advantageous?
  - If we are constantly using the same query to look up information, let's say you are constantly looking what department an employee belongs to, it's beneficial to us to create a index of that table to perform that repetitive query faster.

HTTP/REST Competency

HTTP

- What is HTTP?
  - Hypertext Transfer Protocol. It allows communication between web servers and clients. When you type an URL in a browser and hit enter, your browser sends an HTTP request. The web server responds successfully or an error message.
- What are HTTP verbs?
  - GET
  - POST
  - PUT
  - PATCH
  - DELETE
- What is GET usually used for?
  - Requests data from the server. (Get list of usernames)
- What is POST usually used for?
  - Submits data to the server. Data is put in the response body. (like submitting a form)
- What is PUT usually used for?

    - Updates or creates a resource. The difference is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly may have side effects when creating the same resource multiple times.
- What is PATCH usually used for?
    - To make partial modifications to a resource on the server.
- What is DELETE usually used for?
    - Deletes a resource
- What are 100-level status codes for?
    - Informational response: Means the server has received the request and is processing it.
- What are 200-level status codes for?
    - Successful response: request was successful and server is sending back requested data
- What are 300-level status codes for?
    - Redirection message: the requested resource has been moved to a different URL
- What are 400-level status codes for?
    - Client error response: there was an error in the request or it might not exist
- What are 500-level status codes for?
    - Server error response: error on the server while processing requests
- What is a path parameter?
    - They pass data to and from a server. They are part of the URL path and identify a specific resource on the server like a user or product.
      /cars
- What is a query parameter?
    - A query parameter is added to the end of the URL after a question mark to either filter, sort, or search for specific data in the database.
      /cars?color=blue
- What is a request body?
    - Data that is sent from the client to the server in an HTTP request. An example is when you submit a form to a website.
- What is a response body?
    - Data that is sent from the server to the client in an HTTP response. An example is when you make a request to a web API, the data is returned in the response body.
- What are headers?
    - They provide additional information in an HTTP response like type of data being sent in the response body.
- What is JSON?

    - JavaScript Object Notation is a data interchange format to send and receive data in an HTTP request.

REST

- What is REST?

  REST is an acronym for **RE**presentational **S**tate **T**ransfer and an architectural style for **distributed hypermedia systems**. Roy Fielding first presented it in 2000 in his famous dissertation.

  Like other architectural styles, REST has its guiding principles and constraints. These principles must be satisfied if a service interface needs to be referred to as **RESTful**.

- Why do we use REST?

  REST allows easy evolution of an API design. And that's the key with REST - you're creating an API. Some of the comments have touched on aspects of this thought, but have not actually brought the core issue to life. When you are dealing with REST, you are creating an API that would be used by clients (or yourself). The HTTP actions on the resources give a clear indication to the clients of the API design and functionality. Therefore, when we use the correct HTTP verbs properly, we are declaring an API that is standardised and understandable from a client perspective.

- What is a resource in REST?

  REST resources: XML, text-based, JSON,

- What does it mean to be stateless?

  REST statelessness means being free from the application state.

  EXAMPLE of Statelessness:

  Suppose we have an API where we want to log in and order some goods, the API deployed on many servers can serve many requests, even from the same account without storing the authentication details or provided token state.

- What do we need to do to make an endpoint RESTful?
    - Accept and respond with JSON
    - Use nouns instead of verbs in endpoint paths
    - Name collections with plural nouns
    - Nesting resources for hierarchical objects
    - Handle errors gracefully and return standard error codes
    - Allow filtering, sorting, and pagination
    - Maintain Good Security Practices

    - Cache data to improve performance
    - Versioning our APIs

Javalin

- What is Javalin?
    - A web framework that allows Java to handle HTTP requests as a server and build an API.
- How can I build a server using Javalin?
    - First add a Javalin dependency to your project, and then import Javalin into your project using "import io.javalin.Javalin"
    - Instantiate a Javalin object using Javalin.create() and then start the app by calling the .start() method on that object.
    - You will need to construct API endpoints to handle the client-server HTTP communication.
- How can I design an endpoint in Javalin?
    - Call an HTTP action such as .get() or .post() on the Javalin object. Pass two arguments, a URL path and this::someHandlerName to associate the endpoint with the handler containing the Context object that processes that request.
- What is the Context object for in Javalin?
    - The Context class contains all of the methods needed to handle HTTP requests and the object is commonly used to receive and respond with JSON format data.
- Can you explain the 3-layer controller-service-DAO architecture?
    - The controller layer contains the API that handles HTTP requests, including URL endpoints that receive, transform, and pass data to the service layer. It also is used to send a return response to the client after the request is processed.
    - The service layer contains "business logic" which is another term for filtering or validating user input, like determining if a user provided the information needed to succesfully utilize that input in the DAO layer. Null returns can be used to pass the request back to the Controller layer if the input doesn't follow the "business logic" to preserve database integrity.
    - The DAO layer contains the Data Access Objects that directly communicate with the database by preparing and executing SQL statements using the validated user input from the service layer.