Jyle Darling

M093437

Knowledge Based Questions

AT1

# Questions

**Q1: Describe the Architectural framework for web-enabled application development. Give an example of how this is used in application development.**

Web enabled application architecture describes the interactions between applications, databases, and middleware system operating on the web. It ensures that many different applications can all run simultaneously and still provide the desired functionality in a reasonable time frame.

The application code is split between two groups, Client side code, and Server side code. Server side code must have the ability to respond to HTTP requests, and is responsible for creating the page that a user requests, while also storing many different types of data. This code cannot be seen by the user. Languages like Java, Ruby or PHP work well for servers.

Client side code is parsed by the web browser and can be seen and modified by the user. This code communicates via HTTP requests and cannot read files directly from a server. Languages like CSS HTML and JavaScript are ideal for writing the client side code.

**Q2 Describe the mechanisms for Inter-process communication. For each feature explain why it is needed.**

There are two main types of Inter process communication, either between related processes initiating only one process, like in a parent child process scenario, or between unrelated processes, with two or more different processes. The main mechanisms are:

**Pipes** – these facilitate communication between two related processes and is referred to as a half-duplex mechanism. This means that the first process sends a message to the second process. In order for the second process to communicate back with the first, a second pipe must be used, which can then be described as full duplex.

**FIFO** – is used to communicate between two unrelated processes, this mechanism is full duplex meaning both processes can communicate with each other at the same time.

**Message Queues** – this provides communication between two or more processes and is a full duplex mechanism. The benefit of message queues allows process to passing messages to a que, which can then be retrieved out of the que, once retrieved the message is no longer available.

**Shared Memory** – this is a Communication between two or more processes is achieved by all processes sharing a piece of memory. The shared memory needs to be protected from each other by synchronizing access to all the processes.

**Semaphores** – are used for synchronising access to multiple processes. When one of the processes want to access the memory, it needs to be protected while this is happening, once completed the memory is released. This needs to be repeated by all the processes to ensure the data remains secure.

**Signals** – allow multiple processes to communicate via a source sending a signal, which is then recognised by the destination process and is handled accordingly.

**Q3 In software development for large applications explain two development methodologies contrast these with waterfall and explain why they are more appropriate for large applications.**

The first method which is better suited than waterfall is the Iterative model. It follows the same steps as the waterfall model, but by allowing multiple iterations of the development process, it provides more flexibility during the development process, and is more feasible for changing requirements through this process. It allows the client to be more involved in the development and better steer the course of the application to be meet the needs of the intended users, and be better suited overall.

Similar to the iterative model, a range of agile methods could also be well suited for large scale applications. One framework used is the Large Scale Scrum (LeSS). As it focuses on a scrum process, a sprint is done over a short period, usually a week to a month, with daily meeting occurring during this process. One this sprint is completed, the product is “shipped” and another product feature is selected from the backlog to be worked on in the next sprint.

Both of these methods provide much more versatility and flexibility over the waterfall method. As the client is constantly involved in the development process, less time is spent developing features that miss the mark in terms of the client’s actual requirements. This helps the project to stay on time and on budget, and when this comes to large scale applications, a product being delivered late or over budget has a significant impact on company finances.

**Q4 Describe at least two testing techniques that could be used in distributed application development.**

Depending on the sensitivity of the application itself one method could be alpha and beta testing. Releasing a working version of the application to a small pre-selected group who will provide feedback of the application will allow the developers to get a wide range of users testing the code and sending bug reports back for the development team to evaluate.

Another method would either be white or grey box testing undertaken by the development team, during an iterative development process.

**Q5 Describe at least three design patterns and how they are used.**

**Abstract factory** - this provides an interface for creating families of related or dependent objects without specifying their concrete classes. In normal usage, the client software creates a concrete implementation of the abstract factory and then uses the generic interface of the factory to create the concrete objects that are part of the theme.

**Factory Method** – this defines an interface for creating a single object, but let’s subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses.

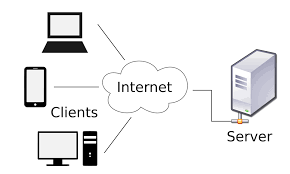
**Iterator** - In object-oriented programming, the iterator pattern is a design pattern in which an iterator is used to traverse a container and access the container's elements. The iterator pattern decouples algorithms from containers; in some cases, algorithms are necessarily container-specific and thus cannot be decoupled.

**Q6 Describe the principles of user interface design, you may use images in your answer.**

Here are eight basic principle of a good UI design.

* **Clear**. Clarity is the most important element of user interface design. Indeed, the whole purpose of user interface design is to enable people to interact with your system by communicating meaning and function. If people can’t figure out how your application works or where to go on your website they’ll get confused and frustrated.
* **Concise**. It is easy to add definitions and explanations, but every time you do that you add mass. Your interface grows. Add too many explanations and your users will have to spend too much time reading through them.
* **Familiar**. Familiar is just that: something which appears like something else you’ve encountered before. When you’re familiar with something, you know how it behaves — you know what to expect. Identify things that are familiar to your users and integrate them into your user interface.
* **Responsive**. Responsive means a couple of things. First of all, responsive means fast. The interface, if not the software behind it, should work fast. Responsive also means the interface provides some form of feedback. The interface should talk back to the user to inform them about what’s happening.
* **Consistent**. Consistent interfaces allow users to develop usage patterns — they’ll learn what the different buttons, tabs, icons and other interface elements look like and will recognize them and realize what they do in different contexts. They’ll also learn how certain things work, and will be able to work out how to operate new features quicker, extrapolating from those previous experiences.
* **Attractive**. When your software is pleasant to use, your customers or staff will not simply be using it — they’ll look forward to using it. There are of course many different types of software and websites, all produced for different markets and audiences. What looks ‘good’ for any one particular audience will vary. This means that you should fashion the look and feel of your interface for your audience.
* **Efficient**. What you really need to do to make an interface efficient is to figure out what exactly the user is trying to achieve, and then let them do exactly that without any fuss. You have to identify how your application should ‘work’.
* **Forgiving.** A forgiving interface is one that can save your users from costly mistakes. For example, if someone deletes an important piece of information, can they easily retrieve it or undo this action?

**Q7 Describe the client server model (you may use images in your answer).**

The client-server model describes how a server provides resources and services to one or more clients. Examples of servers include web servers, mail servers, and file servers. Each of these servers provide resources to client devices, such as desktop computers, laptops, tablets, and smartphones. Most servers have a one-to-many relationship with clients, meaning a single server can provide resources to multiple clients at one time.

**Q8 Explain two dynamic data structures in your answer you must describe how much space and time they take, ensure to reference O notation in your answer.**

One example of a dynamic data structure used in java is a doubly linked list, which is a list of items that have a reference to the item preceding it, and the item following it within the list. The worst case space complexity is O(n), the average and worst case for time complexity relating to Access and search methods are O(n), and the average and worst case time complexity relating to Insertion and Deletion methods are O(1).

Another example is a Stack, a stack follows the basic principle of first in last out. In addition to the basic push and pop operations, the class provides three more functions of empty, search and peek. The worst case space complexity is O(n), the average and worst case for time complexity relating to Access and search methods are O(n), and the average and worst case time complexity relating to Insertion and Deletion methods are O(1).

**Q9 Explain hash functions and data structures using hash functions in your answer you must describe how much space and time they take.**

A hash function is used to map data of arbitrary size (keys) to fixed size values. Values returned from a hash function are referred to as hash values or simply as hashes. The values are used to index a fixed sized table called a hash table, and using a hash function to index a hash table is referred to as hashing. In many cases hash table are more efficient than search trees or similar lookup table structures.

They have a worst and average size complexity of O(n), an average search, insert and delete time complexity of O(1) and a worst case search, inert and delete time complexity of O(n).

**Q10 Explain three sorting algorithms in your answer you must describe how much space and time they take, ensure to reference O notation in your answer.**

**Merge sort** - takes advantage of the ease of merging already sorted lists into a new sorted list. It starts by comparing every two elements (i.e., 1 with 2, then 3 with 4...) and swapping them if the first should come after the second. It then merges each of the resulting lists of two into lists of four, then merges those lists of four, and so on; until at last two lists are merged into the final sorted list. However, it has additional O(n) space complexity, and involves a large number of copies in simple implementations. The best average and worst-case time complexity is Ω(n log(n)).

**Quick sort** - works on the divide-and-conquer principle. First, it partitions the list of items into two sub lists based on a pivot element. All elements in the first sub list are arranged to be smaller than the pivot, while all elements in the second sub list are arranged to be larger than the pivot. The same partitioning and arranging process is performed repeatedly on the resulting sub lists until the whole list of items are sorted. It has a space complexity worst case of O(log(n)), a time complexity best and average of Ω(n log(n)), and a worst case of O(n^2).

**Comb sort** - is a relatively simple sorting algorithm that improves on bubble sort. In bubble sort, when any two elements are compared, they always have a gap (distance from each other) of 1. The basic idea of comb sort is that the gap can be much more than 1. The inner loop of bubble sort, which does the actual swap, is modified such that the gap between swapped elements starts as the length of the array and goes down (for each iteration of outer loop) in steps of a "shrink factor" which is usually 1.3. Worst case size complexity of O(1), time complexity best case Ω(n), average and worst-case O(n^2).

**Q11 Explain three searching algorithms in your answer you must describe how much space and time they take, ensure to reference O notation in your answer.**

**Binary search** – known as a divide and conquer search works by comparing the target value in the middle element of an array, if they are not equal, the half of the array that cannot contain the target value(too low or too high) is eliminated and the search continues on the remaining half, again checking comparing the target to the middle element. This process is repeated until a result is found, or the target value is determined not to be in the list. it has a worst-case time complexity of O(log(n)), however the array must be sorted in order to utilise this search method.

**Linear search** - is a simple search algorithm that checks every record until it finds the target value. Linear search can be done on a linked list, which allows for faster insertion and deletion than an array. Binary search is faster than linear search for sorted arrays except if the array is short, however an array does not need to be sorted in order to utilise a linear search. Worst case time complexity of O(log(n)).

**Hashing** - For implementing associative arrays, hash tables are generally faster than binary search on a sorted array of records. However, hashing is not useful for approximate matches, such as computing the next-smallest, next-largest, and nearest key, as the only information given on a failed search is that the target is not present in any record. They have a average time complexity of O(1) and worst case of O(n).

**Q12 Explain three features of the Java language and three examples of Java syntax. In your answer describe the characteristics of Java.**

Java is an Object Orientated Programming language (OOP), so within java, everything is essentially an object which leads the language to be easily extended via java own contributors, or third parties. Java is platform independent, so when a java application is compiled, it is done so into a platform independent byte code, which is then distributed over the internet and interpreted by the Java Virtual Machine(JVM) on which every program it is being run on. Java is multithreaded which makes it possible to write programs that can perform many tasks simultaneously. This allows developers to construct interactive applications that run smoothly.

The syntax of java is mostly derived from C and C++. An identifier is the name of an element in the code. There are certain standard naming conventions to follow when selecting names for elements and identifiers in Java are case-sensitive. The Java Syntax contains a list of key words that have a pre-determined meaning value or behaviours associated to them, boolean for example is a literal keyword that is used to denote a true or false variable. Variables are identifiers associated with values. They are declared by writing the variable's type and name, and are optionally initialized in the same statement by assigning a value. The separators { and } signify a code block and a new scope. Class members and the body of a method are examples of what can live inside these braces in various contexts.

**Q13 Explain the difference between input and output requirements give an example of each.**

The input and output of a java program can be boiled down into simple requirements, the input requires data to be given to the program, this data is given either by hardcoding into the code itself, or from data sent while the program is running either by user inputs from a keyboard or mouse, or data received from other programs or sensors.

The output is the data that we receive from the program in the form of a result of the code methods. This output may be displayed to the user via the monitor, or by another signal being sent to another process or method. The data in both case is often referred to as an input stream, or output stream.

**Q14 Describe two software development methodologies compare them to waterfall and explain the differences between them (you may use images in your answer).**

The first method is the Iterative model. It follows the same steps as the waterfall model, but by allowing multiple iterations of the development process, it provides more flexibility during the development process, and is more feasible for changing requirements through this process. It allows the client to be more involved in the development and better steer the course of the application to be meet the needs of the intended users, and be better suited overall.

Similar to the iterative model, another method would be SCRUM, the scrum process is focused on individual sprints, done over a short period, usually a week to a month, with daily meetings occurring during this process. One this sprint is completed, the product is “shipped” and another product feature is selected from the backlog to be worked on in the next sprint.

Both of these methods provide much more versatility and flexibility over the waterfall method. As the client is constantly involved in the development process, less time is spent developing features that miss the mark in terms of the client’s actual requirements. This helps the project to stay on time and on budget.

**Q15 Describe the system layers, you must include in your answer the data network, hardware, operating system, database management systems, web servers, application servers, and client deployment.**

Hardware refers to physical objects used by people to interact with the computer software, things like pc’s and their internal components, laptops, tablets, keyboards, routers and modems are all considered hardware.

A data network is used to describe a series or collection of connected hardware, enabling the hardware devices to communicate and exchange data with each other over local networks, or via the internet. Specific hardware used in data networks are routers, modems, servers, switches, as well as computers themselves to process the exchanged data.

Operating systems are the main piece of software that we interact with when using electronic devices. A large number of computers run on a Windows operating system, while iPhone’s run on a version of the IOS (iPhone Operating System).

Database management systems refers to the software that enables users to directly interact with a database and its contents. The program PhpMyAdmin is one such program that helps user create edit and remove database’s, database tables or contents within a database table.

Web servers are a dedicated server or group of servers designed for storing website information. Clients are able to request information from these servers via http requests via web browsers, which will then communicate with their web browser and show them the stored website information.

Application servers are similar to web servers in that they are a dedicated server designed to store and run application data. Because the applications being run would typically contain more information than a standard web page, these servers must be configured to handle a larger throughput of information in a timely manner.

Client deployment refers to the process of transferring or installing a completed software package to a client or a client’s company system. This would generally occur at the end of a software development process, either by direct installation, or through a website or server. Following this process the development team would monitor the installed system and make any changes or bug fixes through continue maintenance of the software.

**Q16 Explain the techniques and processes related to small size application development and how they are different from large sized application development.**

For the most part the basic techniques and processes are the same for both small and large size applications, the only difference between them is the scale. An iterative development method is most often time better than a rigid method like waterfall. The cycle or requirement gathering, design, implantation, testing and demonstration remains the same, however the number of individual cycles will vary depending on the application size and complexity. Understanding the business domain will always be a solid technique to help reduce the scale or the requirements needed throughout the development. Likewise keeping the client involve with every iteration of the development keeps the project from straying too far away from the original concept or design, whilst ensuring the client that progress is being made constantly. testing is another process that remains the same for both application sizes, but once again the scale of the testing determined by the application size.