1. *4 principle of OOP*

Encapsulation

Encapsulation is the mechanism of hiding of data implementation by restricting access to public methods.

Abstraction

Abstract means a concept or an Idea which is not associated with any particular instance. Using abstract class/interface we express the intent of the class rather than the actual implementation. In a way, one class should not know the inner details of another in order to use it, just knowing the interfaces should be good enough.

Inheritance

Inheritances expresses "is a" relationship between two objects. Using proper inheritance, In derived classes we can reuse the code of existing super classes.

Polymorphism

It means one name many forms. It is further of two types - static and dynamic. Static polymorphism is achieved using method overloading and dynamic polymorphism using method overriding.

What is aggregation, how is it different from composition ?

Both of these are special type of association and differ only in weight of relationship. Composition is stronger form of "is part of" relationship compared to aggregation "has a". In composition, the member object can not exist outside the enclosing class while same is not true for Aggregation.

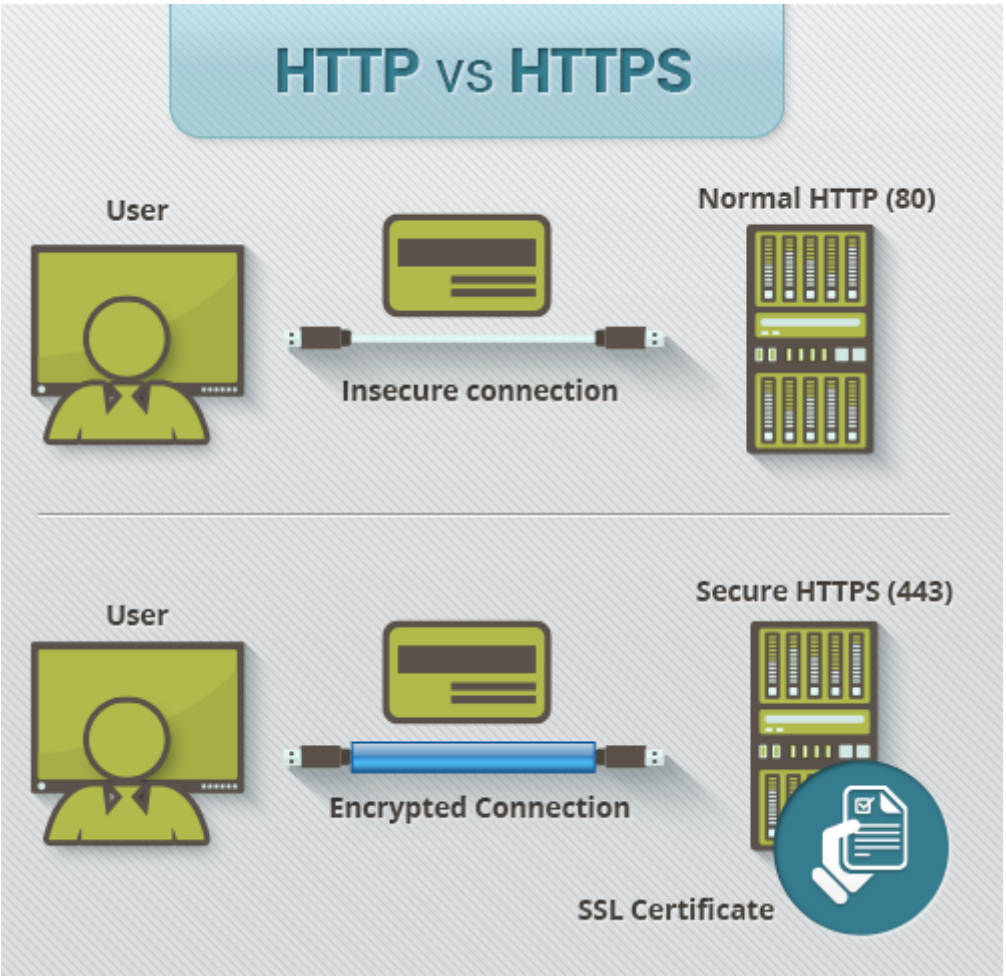
2. *Tell me about a project where you used the principles of OOP.*

Cruise, employee: captain, sailor, cooker, their salary; customer, order, dishes….

3. Past internship experiences
4. What is the importance of technology to JP Morgan?
5. Describe a time you worked with a difficult team member and how you handled it.
6. Given a 7 day prediction of a single stock price, write an algorithm to maximize posits from buying/selling given a daily investment cap.
7.

```
public int maxProfit(int[] prices) {
                if (prices.length == 0) {
                        return 0 ;
                }
                int max = 0 ;
                int sofarMin = prices[0] ;
            for (int i = 0 ; i < prices.length ; ++i) {
                if (prices[i] > sofarMin) {
                        max = Math.max(max, prices[i] - sofarMin) ;
                } else{
                       sofarMin = prices[i];
                }
            }
           return  max ;
        }
```

8. Given an nxm matrix of integers and unlimited physical resources (memory, processor speed), write a fast algorithm to calculate the sum of all matrix member
9. How is https more secure than http



HTTPS pages typically use one of two secure protocols to encrypt communications - SSL (Secure Sockets Layer) or TLS (Transport Layer Security). Both the TLS and SSL protocols use what is known as an 'asymmetric' Public Key Infrastructure (PKI) system. An asymmetric system uses two 'keys' to encrypt communications, a 'public' key and a 'private' key. Anything encrypted with the public key can only be decrypted by the private key and vice-versa.

As the names suggest, the 'private' key should be kept strictly protected and should only be accessible the owner of the private key. In the case of a website, the private key remains securely ensconced on the web server. Conversely, the public key is intended to be distributed to anybody and everybody that needs to be able to decrypt information that was encrypted with the private key.

10. Describe the difference between agile vs waterfall

Answer:

Waterfall emphasizes continuously moving forward. Not as flexible as Agile in terms of changing specs mid-development. Does not allow going back to fix bug/change specs. Agile is more flexible and welcoming of change. Good communication is necessary between developers and clients across all level of the project for Agile to work

11. the difference between Java and Javascript and what a hash table is
    Key **differences between Java and JavaScript**: **Java** is an OOP programming language while **Java** Script is an OOP scripting language. **Java** creates applications that run in a virtual machine or browser while **JavaScript** code is run on a browser only. **Java** code needs to be compiled while **JavaScript** code are all in text.
12. sort algorithm
13. questions on collections, difference between abstract and interface
    1.Main **difference** is methods of a **Java interface** are implicitly **abstract** and cannot have implementations. A **Java abstract class** can have instance methods that implements a default behavior.
    2.Variables declared in a **Java interface** is by default final. An **abstract class** may contain non-final variables.


java.util

Class Collections

java.lang.Object
└ java.util.Collections

---

java.util

Interface Collection<E>

All Superinterfaces:
Iterable<E>

All Known Subinterfaces:
BeanContext, BeanContextServices, BlockingDeque<E>, BlockingQueue<E>, Deque<E>, List<E>, NavigableSet<E>, Queue<E>, Set<E>, SortedSet<E>

All Known Implementing Classes:
AbstractCollection, AbstractList, AbstractQueue, AbstractSequentialList, AbstractSet, ArrayBlockingQueue, ArrayDeque, ArrayList, AttributeList, BeanContextServicesSupport, BeanContextSupport, ConcurrentLinkedQueue, ConcurrentSkipListSet, CopyOnWriteArrayList, CopyOnWriteArraySet, DelayQueue, EnumSet, HashSet, JobStateReasons, LinkedBlockingDeque, LinkedBlockingQueue, LinkedHashSet, LinkedList, PriorityBlockingQueue, PriorityQueue, RoleList, RoleUnresolvedList, Stack, SynchronousQueue, TreeSet, Vector

14. Describe Bubblesort
15. What algorithms do you know?
16. Why do you like your language?
17. About my past projects and my interest. Asked question about data structure.

18. What is the difference between method overload and override?
19. Tell me a brief summer of the coding techniques you know.
20. Know the four fundamental OOP concepts and be able to answer applied questions on them.
21. Is Java pass by reference or pass by value?
22. Explain the efficiency of garbage collection in Java and C.
23. How to make a looping program more efficient.
24. *For a startup company which would be the best type of cloud* ***computing model for it to adopt***

    SAAS