Interview Process:

Introduce yourself .. (and all that you read before)

- Read about the interview process and what will happens in the interview .
- Intro includes about yourself and your path so far (experience, past projects, personal projects, biggest challenges..)
- Go deep on past projects: implementation options, what were the biggest things that were learned and what would do differently

Let's First start with some puzzles questions which can be or cannot be neglected depend on your interview.

****

Puzzles Questions:

#1 Bag of Coins

You have 10 bags full of coins. In each bag are infinite coins. But one bag is full of forgeries, and you can't remember which one. But you do know that a genuine coins weigh 1 gram, but forgeries weigh 1.1 grams. You have to identify that bag in minimum readings. You are provided with a digital weighing machine.

Answer: 1 reading.

Take 1 coin from the first bag, 2 coins from the second bag, 3 coins from the third bag and so on. Eventually, we'll get 55 (1+2+3…+9+10) coins. Now, weigh all the 55 coins together. Depending

on the resulting weighing machine reading, you can find which bag has the forged coins such that if the reading ends with 0.4 then it is the 4th bag, if it ends with 0.7 then it is the 7th bag and so on.

#2 Prisoners and hats

There are 100 prisoners all sentenced to death. One night before the execution, the warden gives them a chance to live if they all work on a strategy together. The execution scenario is as follows –

On the day of execution, all the prisoners will be made to stand in a straight line such that one prisoner stands just behind another and so on. All prisoners will be wearing a hat either of Blue colour or Red. The prisoners don't know what colour of hat they are wearing. The prisoner who is standing at the last can see all the prisoners in front of him (and what colour of hat they are wearing). A prisoner can see all the hats in front of him. The prisoner who is standing in the front of the line cannot see anything.

The executioner will ask each prisoner what colour of hat they are wearing one by one, starting from the last in the line. The prisoner can only speak "Red" or "Blue". He cannot say anything else. If he gets it right, he lives otherwise he is shot instantly. All the prisoners standing in front of him can hear the answers and gunshots.

Assuming that the prisoners are intelligent and would stick to the plan, what strategy would the prisoners make over the night to minimize the number of deaths?

Answer:

The strategy is that the last person will say 'red' if the number of red hats in front of him are odd and 'blue' if the number of red hats in front of him are even. Now, the 99th guy will see the if the red hats in front of him are odd or even. If it is odd then obviously the hat above him is blue, else it is red. From now on, it's pretty intuitive.

#3 Blind games

You are in a dark room where a table is kept. There are 50 coins placed on the table, out of which 10 coins are showing tails and 40 coins are showing heads. The task is to divide this set of 50 coins into 2 groups (not necessarily same size) such that both groups have same number of coins showing the tails.

Answer:

Divide the group into two groups of 40 coins and 10 coins. Flip all coins of the group with 10 coins.

#4 Sand timers

You have two sand timers, which can show 4 minutes and 7 minutes respectively. Use both the sand timers(at a time or one after other or any other combination) and measure a time of 9 minutes.

Answer:

1. Start the 7 minute sand timer and the 4 minute sand timer.
2. Once the 4 minute sand timer ends turn it upside down instantly.
3. Once the 7 minute sand timer ends turn it upside down instantly.
4. After the 4 minute sand timer ends turn the 7 minute sand timer upside down(it has now minute of sand in it)

So effectively 8 + 1 = 9.

#5 Chaos in the bus

There is a bus with 100 labeled seats (labeled from 1 to 100). There are 100 persons standing in a queue. Persons are also labelled from 1 to 100.

People board on the bus in sequence from 1 to n. The rule is, if person ‘i’ boards the bus, he checks if seat ‘i’ is empty. If it is empty, he sits there, else he randomly picks an empty seat and sit there. Given that 1st person picks seat randomly, find the probability that 100th person sits on his place i.e. 100th seat.

Answer:

The final answer is the probability that the last person ends in up in his proper seat is exactly 1/2

The reasoning goes as follows:

First, observe that the fate of the last person is determined the moment either the first or the last seat is selected! This is because the last person will either get the first seat or the last seat. Any other seat will necessarily be taken by the time the last guy gets to ‘choose’.

Since at each choice step, the first or last is equally probable to be taken, the last person will get either the first or last with equal probability: 1/2.

#6 Mad men in a circle

N persons are standing in a circle. They are labelled from 1 to N in clockwise order. Every one of them is holding a gun and can shoot a person on his left. Starting from person 1, they starts shooting in order e.g for N=100, person 1 shoots person 2, then person 3 shoots person 4, then person 5 shoots person 6……..then person 99 shoots person 100, then person 1 shoots person 3, then person 5 shoots person 7……and it continues till all are dead except one. What's the index of that last person ?

Answer:

Write 100 in binary, which is 1100100 and take the complement which is 11011 and it is 27. Subtract the complement from the original number. So 100 – 27 = 73.

Try it out for 50 people. 50 = 110010 in binary.

Complement is 1101 = 13. Therefore, 50 – 13 = 37.

For the number in form 2^n, it will be the first person. Let's take an example:

64 = 1000000

Complement = 111111 = 63.

64-63 = 1.

You can apply this for any 'n'.

#7 Lazy people need to be smart

Four glasses are placed on the corners of a square Lazy Susan (a square plate which can rotate about it's center). Some of the glasses are upright (up) and some upside-down (down).

A blindfolded person is seated next to the Lazy Susan and is required to re-arrange the glasses so that they are all up or all down, either arrangement being acceptable (which will be signalled by say ringing of a bell).

The glasses may be rearranged in turns with subject to the following rules: Any two glasses may be inspected in one turn and after feeling their orientation the person may reverse the

orientation of either, neither or both glasses. After each turn the Lazy Susan is rotated through a random angle.

The puzzle is to devise an algorithm which allows the blindfolded person to ensure that all glasses have the same orientation (either up or down) in a finite number of turns. (The algorithm must be deterministic, i.e. non-probabilistic)

Answer:

This algorithm guarantees that the bell will ring in at most five turns:

5. On the first turn, choose a diagonally opposite pair of glasses and turn both glasses up.
6. On the second turn, choose two adjacent glasses at least one will be up as a result of the previous step. If the other is down, turn it up as well. If the bell does not ring, then there are now three glasses up and one down.
7. On the third turn, choose a diagonally opposite pair of glasses. If one is down, turn it up and the bell will ring. If both are up, turn one down. There are now two glasses down, and they must be adjacent.
8. On the fourth turn, choose two adjacent glasses and reverse both. If both were in the same orientation then the bell will ring. Otherwise there are now two glasses down and they must be diagonally opposite.
9. On the fifth turn, choose a diagonally opposite pair of glasses and reverse both. The bell will ring.

#8 These kids deserve medals

There are 10 incredibly smart boys at school: A, B, C, D, E, F, G, H, I and Sam. They run into class laughing at 8:58 am, just two minutes before the playtime ends and are stopped by a stern looking teacher: Mr Rabbit.

Mr Rabbit sees that A, B, C and D have mud on their faces. He, being a teacher who thinks that his viewpoint is always correct and acts only to enforce rules rather than thinking about the world that should be, lashes out at the poor kids.

“Silence!”, he shouts. “Nobody will talk. All of you who have mud on your faces, get out of the class!”. The kids look at each other. Each kid could see whether the other kids had mud on their faces, but could not see his own face. Nobody goes out of the class.

“I said, all of you who have mud on your faces, get out of the class!”

Still nobody leaves. After trying 5 more times, the bell rings at 9 and Mr Rabbit exasperatedly yells: “I can clearly see that at least one of you kids has mud on his face!”.

The kids grin, knowing that their ordeal will be over soon. Sure enough, after a few more times bawling of “All of you who have mud on your faces, get out of the class!”, A, B, C and D walk out of the class.

Explain how A, B, C and D knew that they had mud on their faces. What made the kids grin? Everybody knew that there was at least one kid with mud on his face. Support with a logical statement that a kid did not know before Mr Rabbit’s exasperated yell at 9, but that the kid knew right after it.

Answer:

After Mr Rabbit’s first shout, they understood that at least one boy has mud on his face. So, if it was exactly one boy, then the boy would know that he had mud on his face and go out after one shouting.

Since nobody went out after one shouting, they understood that at least two boys have mud on their faces. If it were exactly two boys, those boys would know (they would see only one other’s muddy face and they’d understand their face is muddy too) and go out after the next shouting.

Since nobody went out after the second shouting, it means there are atleast three muddy faces And so on, after the fourth shouting, A, B, C and D would go out of the class.

This explanation does leave some questions open. Everybody knew at least three others had mud on their faces, why did they have to wait for Mr. Rabbit’s shout at the first place? Why did they have to go through the all four shoutings after that as well?

In multi-agent reasoning, an important concept arises of common knowledge. Everybody knows that there are at least three muddy faces but they cannot act together on that information without knowing that everybody else knows that too. And that everybody knows that everybody knows that and so on. This is what we’ll be analyzing. It requires some imagination, so be prepared.

A knows that B, C and D have mud on their faces. A does not know if B knows that three people have mud on their faces. A knows that B knows that two people have mud on their faces. But A can’t expect people to act on that information because A does not know if B knows that C knows that there are two people with mud on their faces. If you think this is all uselessly complicated, consider this:

A can imagine a world in which he does not have mud on his face. (Call this world A) In A's world, A can imagine B having a world where both A and B do not have mud on their faces. (Call this world AB)

A can imagine a world where B imagines that C imagines that D imagines that nobody has mud on their faces. (Call this world ABCD). So when Mr Rabbit shouted initially, it could have been that nobody was going out because a world ABCD was possible in which nobody should be going out anyway.

So here's a statement that changes after Mr. Rabbit's yell. World ABCD is not possible i.e. A cannot imagine a world where B imagines that C imagines that D imagines that nobody has mud on their faces. So now in world ABC, D knows he has mud on his face. And in world ABD, C knows he has mud on his face and so on.

#9 More prisoners and more hats

There are 7 prisoners sitting in a circle. The warden has caps of 7 different colours (an infinite supply of each colour). The warden places a cap on each prisoner's head – he can chose to place any cap on any other's head. Each prisoner can see all caps but her/his own. The warden orders everybody to shout out the colour of their respective caps simultaneously. If any one is able to guess her/his colour correctly, he sets them free. Otherwise, he send them in a dungeon to rot and die. Is it possible to devise a scheme to guarantee that nobody dies?

Answer:

Assign to each of the 7 colours a unique number from 0-6. Henceforth, we will only be doing modular arithmetic(modulo 7).

Assign to each of the 7 prisoners a unique number from 0-6. If the number assigned to prisoner P is N, then P always guesses that the sum of the colours assigned to all prisoners is M (modulo 7). Thus, he calculates his own colour under this assumption (`= (M - sum(colours of the 6 prisoners he can see))%7`).

There will always be a prisoner who guesses the correct sum (as the sum lies in 0-6), and this prisoner therefore correctly guesses his own colour.

If there is a solution, then exactly one prisoner is correct (no more). This is because there are 7^7 scenarios.

Each prisoner's response is a function of the colours of the other 6, so if you fix their colours and vary his colour, you can see that he will be correct in exactly one-seventh of the cases (=7^6). The sum (across all scenarios) of the number of prisoners who are correct is 7*(7^6)=7^7.

If each scenario is to have at least one person right, this implies that each scenario cannot have more than one person who is right.

Being right about one's colour is equivalent to being right about the sum of colours of all prisoners (modulo 7). (The colours of the other 6 are known.) So guessing one's colour is the same as guessing the sum. How do we make sure that at least one person guesses the correct sum? By making sure that everybody guesses a different sum.

#10 All men must die

One day, an alien comes to Earth. Every day, each alien does one of four things, each with equal probability to:

(i) Kill himself
(ii) Do nothing
(iii) Split himself into two aliens (while killing himself)
(iv) split himself into three aliens (while killing himself)

What is the probability that the alien species eventually dies out entirely?

Answer:

The answer is √2 – 1.

Suppose that the probability of aliens eventually dying out is x.

Then for n aliens, the probability of eventually dying out is xn because we consider every alien as a separate colony. Now, if we compare aliens before and after the first day, we get:

$$x = (1/4) * 1 + (1/4) * x + (1/4) * x^2 + (1/4) * x^3$$

$$x^3 + x^2 - 3x + 1 = 0$$

$$(x - 1)(x\ 2 + 2x - 1) = 0$$

We get, $x = 1, -1 - \sqrt{2}$, or $-1 + \sqrt{2}$

We claim that x cannot be 1, which would mean that all aliens eventually die out. The number of aliens in the colony is, on average, multiplied by 0+1+2+3 4 = 1.5 every minute, which means in general the aliens do not die out. (A more rigorous line of reasoning is included below.) Because x is not negative, the only valid solution is $x = \sqrt{2} - 1$.

To show that x cannot be 1, we show that it is at most $\sqrt{2}-1$.

Let xn be the probability that a colony of one bacteria will die out after at most n minutes. Then, we get the relation:

$$x_n + 1 = 1/4\ (1 + x_n + x^2 n + x^2 n)$$

We claim that $x_n \leq \sqrt{2} - 1$ for all n, which we will prove using induction.

It is clear that $x_1 = 1/4 \leq \sqrt{2} - 1$. Now, assume $x_k \leq \sqrt{2} - 1$ for some k. We have:

$$x_{k+1} \leq 1/4\ (1 + x_k + x^2 k + x^3 k\ )$$

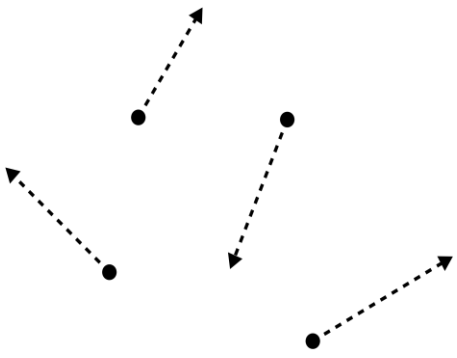
$$\leq 1/4\ (\ 1 + (\sqrt{2} - 1) + (\sqrt{2} - 1)^2 + (\sqrt{2} - 1)^3\ )$$

$$= \sqrt{2} - 1$$

which completes the proof that $x_n \leq \sqrt{2} - 1$ for all n. Now, we note that as n becomes large, x_n approaches x. Using formal notation, this is:

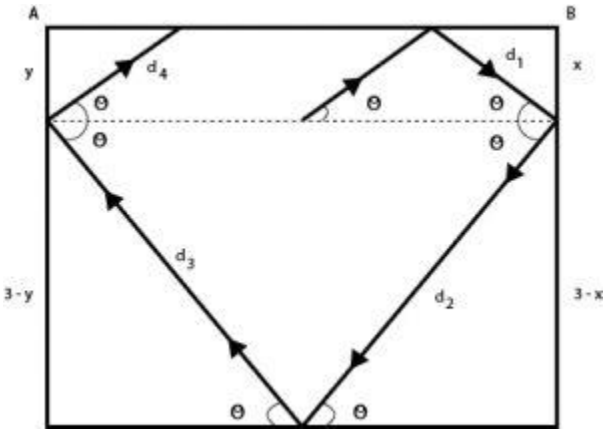
$x = \lim_{n \to \infty} x_n \leq \sqrt{2} - 1$, so x cannot be 1.

#11 Lumos



A photon starts moving in random direction from the center of square of size 3. Let's say it first colloids to the glass wall AB. What is the expected distance traveled by photon before hitting the wall AB again?

Answer:



Above is a pictorial representation of the photon. We can calculate it's distance as shown below:

$d1 = x \operatorname{cosec}(\theta)$

$d2 = (3 - x) \operatorname{cosec}(\theta)$

$d3 = (3 - y) \operatorname{cosec}(\theta)$

$d4 = y \operatorname{cosec}(\theta)$

Total distance = $d1 + d2 + d3 + d4$

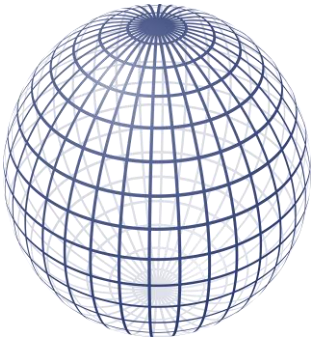
$= 6 \operatorname{cosec}(\theta)$

We know, Θ varies between Π/4 and 3Π/4

Therefore, $E(\text{distance}) = 6 \, E(\operatorname{cosec}(\theta)) = 6 \times (2/\Pi) \int \operatorname{cosec}(\theta) d\theta$ (limits Π/4 to 3Π/4)

$= 12/\Pi \ln(\sqrt{2} + 1/\sqrt{2} + 1)$

#12 4 points in a sphere



Consider a unit sphere. 4 points are randomly chosen on it, what is the probability that the centre (of sphere) lies within the tetrahedron (/ polygon) formed by those 4 points?

Answer:

Let A, B and C be random points on the sphere with Aa, Bb and Cc being diameters.

The spherical (minor) triangle abc is common to the hemispheres abc, bca and cab (where the notation abc represents the hemisphere cut off by the great circle through a and b and containing the point c, etc), therefore the probability that a further random point, D, lies on this triangle is:

`1/2 x 1/2 x 1/2 = 1/8`

(For centre to lie in the tetrahedron D should lie in the triangle i.e the opposite hemisphere of ABC)

#13 Misogynist country

In a country in which people only want boys, every family continues to have children until they have a boy. If they have a girl, they have another child. If they have a boy, they stop. What is the proportion of boys to girls in the country?

Answer:

Following is the required calculation:

Expected Number of boys for 1 family = `1*(Probability of 1 boy) + 1*(Probability of 1 girl and a boy) + 1*(Probability of 2 girls and a boy) + …`

For C couples = `1*(C*1/2) + 1*(C*1/2*1/2) + 1*(C*1/2*1/2*1/2) + …`

Expected Number of boys = `C/2 + C/4 + C/8 + C/16 + …`

Expected Number of boys = `C`

Expected Number of girls for 1 family = `0*(Probability of 0 girls) + 1*(Probability of 1 girl and a boy) + 2*(Probability of 2 girls and a boy) + …`

For C couples = `0*(C*1/2) + 1*(C*1/2*1/2) + 2*(C*1/2*1/2*1/2) + …`

Expected Number of girls = `0 + C/4 + 2*C/8 + 3*C/16 + …`

Expected Number of girls = `C`

Therefore, the proportion is C/C = `1:1`

#14 The Red wedding

A bad king has a cellar of 1000 bottles of delightful and very expensive wine. A neighbour queen plots to kill the bad king and sends a servant to poison the wine.

Fortunately (or say unfortunately) the bad king's guards catch the servant after he could poison only one bottle. Alas, the guards don't know which bottle, but know that the poison is so strong that even if diluted 100,000 times it would still kill the king.

Furthermore, it takes one month to have an effect. The bad king decides he will get some of the prisoners in his vast dungeons to drink the wine. Being a clever bad king, he knows that he needs to murder no more than 10 prisoners – believing he can fob off such a low death rate – and will still be able to drink the rest of the wine (999 bottles) at his wedding party in 5 weeks time.

Explain what is in mind of the king, how will he be able to do so ? (he has only 10 prisoners in his prisons)

Answer:

The number the bottles are 1 to 1000. Now, write the number in binary format. We can write it as:

bottle 1 = 0000000001 (10 digit binary)

bottle 2 = 0000000010

.

.

.

bottle 500 = 0111110100

bottle 1000 = 1111101000

Now, take 10 prisoners and number them 1 to 10. Let prisoner 1 take a sip from every bottle that has a 1 in its least significant bit. And, this process will continue for every prisoner until the last prisoner is reached. For example:

Prisoner = 10 9 8 7 6 5 4 3 2 1

Bottle 924 = 1 1 1 0 0 1 1 1 0 0

For instance, bottle no. 924 would be sipped by 10,9,8,5,4 and 3. That way if bottle no. 924 was the poisoned one, only those prisoners would die.

After four weeks, line the prisoners up in their bit order and read each living prisoner as a 0 bit and each dead prisoner as a 1 bit. The number that you get is the bottle of wine that was poisoned. We know, 1000 is less than 1024 (2^10). Therefore, if there were 1024 or more bottles of wine it would take more than 10 prisoners.

#15 Life and luck

You and your friend are caught by gangsters and made to play a game to determine if you should live or die. The game is simple.

There is a deck of cards and you both have to choose a card. You can look at each other's cards but not at the card you have chosen. You both will survive if both are correct in guessing the card they have chosen. Otherwise both die.

What is the probability of you surviving if you and your friend play the game optimally?

Answer:

We know, A and B have picked a card at random from a deck. A can see B's card and vice versa. So, A knows (s)he has not picked B's card, but apart from that, (s)he knows that the card is equally probable to be any of the other 51 cards. So, if A guesses B's card, they lose. But if A guesses any other card, there's a 1/51 chance that A is right. This also implies that total probability of success <= 1/51.

A's aim now is to tell any card apart from B's card that gives B the most information about B's own card. So they can plan beforehand as follows:

Consider the sequence of cards Clubs 1-13, Diamonds 1-13, Hearts 1-13, Spades 1-13. A will tell the card after B's card in this sequence. (If A says 4 of Hearts, it means B has 3 of Hearts. If A says Ace of Clubs, it means B has King of Spades)

With A's guess, which is always different from B's card, B gets to know exactly which card (s)he has and can always guess correctly. So the probability of success is 1/51, which is the maximum achievable.

#16 Weighing balls

You have 12 balls that all weigh the same except one, which is either slightly lighter or slightly heavier. The only tool you have is a balance scale that can only tell you which side is heavier. Using only three weightings, how can you deduce, without a shadow of a doubt, which is the odd one out, and if it is heavier or lighter than the others?

Answer:

First we weigh {1,2,3,4} on the left and {5,6,7,8} on the right. There are three scenarios which can arise from this:

If they balance, then we know 9, 10, 11 or 12 is fake. Weigh {8, 9} and {10, 11} (Note: 8 is surely not fake). If they balance, we know 12 is the fake one. Just weigh it with any other ball and figure out if it is lighter or heavier.

If {8, 9} is heavier, then either 9 is heavy or 10 is light or 11 is light. Weigh {10} and {11}. If they balance, 9 is fake (heavier). If they don't balance then whichever one is lighter is fake (lighter).

If {8, 9} is lighter, then either 9 is light or 10 is heavy or 11 is heavy. Weigh {10} and {11}. If they balance, 9 is fake (lighter). If they don't balance then whichever one is heavier is fake (heavier).

If {1,2,3,4} is heavier, we know either one of {1,2,3,4} heavier or one of {5,6,7,8} is lighter but it is guarantees that {9,10,11,12} are not fake. This is where it gets really tricky, watch carefully. Weigh {1,2,5} and {3,6,9} (Note: 9 is surely not fake).

If they balance, then either 4 is heavy or 7 is light or 8 is light. Following the last step from the previous case, we weigh {7} and {8}. If they balance, 4 is fake(heavier). If they don't balance then whichever one is lighter is fake (lighter).

If {1,2,5} is heavier, then either 1 is heavy or 2 is heavy or 6 is light. Weigh {1} and {2}. If they balance, 6 is fake (lighter). If they don't balance then whichever one is heavier is fake (heavier).

If {3,6,9} is heavier, then either 3 is heavy or 5 is light. Weigh {5} and {9}. They won't balance. If {5} is lighter, 5 is fake (lighter). If they balance, 3 is fake (heavier).

If {5,6,7,8} is heavier, it is the same situation as if {1,2,3,4} was heavier. Just perform the same steps using 5,6,7 and 8. Unless maybe you are too lazy to try and reprocess the steps, then you continue reading the solution. Weigh {5,6,1} and {7,2,9} (Note: 9 is surely not fake).

If they balance, then either 8 is heavy or 3 is light or 4 is light. Following the last step from the previous case, we weigh {3} and {4}. If they balance, 8 is fake(heavier). If they don't balance then whichever one is lighter is fake (lighter).

If {5,6,1} is heavier, then either 5 is heavy or 6 is heavy or 2 is light. Weigh {5} and {6}. If they balance, 2 is fake (lighter). If they don't balance then whichever one is heavier is fake (heavier).

If {7,2,9} is heavier, then either 7 is heavy or 1 is light. Weigh {1} and {9}. If they balance, 7 is fake (heavier). If they don't balance then 1 is fake (lighter).

#17 Bias and unbias

Robin and Williams are playing a game. An unbiased coin is tossed repeatedly. Robin wins as soon as the sequence of tosses HHT appears. Williams wins as soon as the sequence of tosses HTH appears. The game ends when one of them wins. What are the probabilities of winning for each player?

Answer: (Robin) HHT – **2/3** (Williams) HTH – **1/3**

Let the probability of Robin winning be p. The probability of Williams winning is (1-p). If the first toss is tails, it is as good as the game has not started, hence the probability of Robin winning is p after the first tail.

```
p = (1/2)*p + ….
```

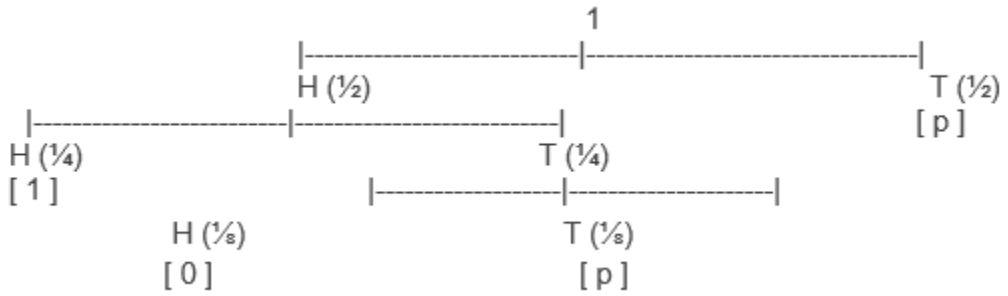
Let the first toss be heads. If the second toss is heads, then Robin definitely wins. Since HH has occurred, and at some point, tails will occur, so HHT will occur. Hence Robin wins with probability 1 for HH.

```
p = (1/2 )*p + (1/2)*((1/2)*1 + .....)
```

Let the second toss be tails. If the third toss is heads, Robin loses as HTH occurs. If the third toss is tails (HTT) – since two tails have occurred in a row, now it is as good as the game has started from the beginning, so the chances of Robin winning are back to p.

T HH HTH HTT

```
p = (1/2)*p + 1/2 ((1/2)*1 + 1/2 ((1/2)*0 + (1/2) * p))
```



p = (1/2)*p + (1/4)*1 + (1/8)*0 + (1/8)*p

Finally, solving this equation gives us p = 2/3.

#18 Chameleons go on a date

On an island live 13 purple, 15 yellow and 17 maroon chameleons. When two chameleons of different colors meet, they both change into the third color. Is there a sequence of pairwise meetings after which all chameleons have the same color?

Answer:

Let <p, y, m> denote a population of p purple, y yellow and m maroon chameleons. Can population <13, 15, 17> be transformed into <45, 0, 0> or <0, 45, 0> or <0, 0, 45> through a series of pairwise meetings?

We can define function:

```
X(p, y, m) = (0p + 1y + 2m) mod 3
```

An interesting property of X is that its value does not change after any pairwise meeting because

```
X(p, y, m) = X(p-1, y-1, m+2) = X(p-1, y+2, m-1) = X(p+2, y-1, m-1)
```

Now X(13, 15, 17) equals 1. However,

```
X(45, 0, 0) = X(0, 45, 0) = X(0, 0, 45) = 0**
```

This means that there is no sequence of pairwise meetings after which all chameleons will have identical colour.

#19 The Einstein puzzle

Answer the question using the given information and hints.

10. In a street there are five houses, painted five different colors.
11. In each house lives a person of different nationality
12. These five homeowners each drink a different kind of beverage, smoke different brand of cigar and keep a different pet.

The Question: Who owns the fish ?

Hints:

13. The British man lives in a red house.
14. The Swedish man keeps dogs as pets.
15. The Danish man drinks tea.
16. The Green house is next to, and on the left of the White house.
17. The owner of the Green house drinks coffee.
18. The person who smokes Pall Mall rears birds.
19. The owner of the Yellow house smokes Dunhill.
20. The man living in the center house drinks milk.
21. The Norwegian lives in the first house.
22. The man who smokes Blends lives next to the one who keeps cats.
23. The man who keeps horses lives next to the man who smokes Dunhill.

24. The man who smokes Blue Master drinks beer.
25. The German smokes Prince.
26. The Norwegian lives next to the blue house.
27. The Blends smoker lives next to the one who drinks water.

FYI – This question is famously known as Einstein Puzzle.

********** Data Structures **********

Data Structures:

- **How do you find the missing number in a given integer array of 1 to 100?**

Ans:
```
private static int getMissingNumber(int[] numbers, int totalCount)
{        int expectedSum = totalCount * ((totalCount + 1) / 2);
        int actualSum = 0;
        for (int i : numbers) {            actualSum += i;        }
return expectedSum - actualSum;
}
```

- **How do you find the duplicate number on a given integer array?**

Ans:
```
static void printRepeating(int arr[], int size)
{
    int i, j;
    System.out.println("Repeated Elements are :");
    for (i = 0; i < size; i++)
    {
        for (j = i + 1; j < size; j++)
```

```
            {
                if (arr[i] == arr[j])
                    System.out.print(arr[i] + " ");
            }
        }
    }
```

- **How do you find the largest and smallest number in an unsorted integer array?**

Ans:

```
public static void largestAndSmallest(int[] numbers) {
    int largest = Integer.MIN_VALUE;
    int smallest = Integer.MAX_VALUE;
    for (int number : numbers) {
        if (number > largest) {
            largest = number;
        } else if (number < smallest) {
            smallest = number;
        }
    }
    System.out.println("Given integer array : " +
Arrays.toString(numbers));
    System.out.println("Largest number in array is : " + largest);
    System.out.println("Smallest number in array is : " + smallest);
}
```

- **How are duplicates removed from a given array in Java? (solution)**
- **How is an integer array sorted in place using the quicksort algorithm? (solution)**
- **How do you remove duplicates from an array in place? (solution)**
- **How do you reverse an array in place in Java? (solution)**
- **How are duplicates removed from an array without using any library? (solution)**

(LinkedList)

- **How do you find the middle element of a singly linked list in one pass? (solution)**
- **How do you check if a given linked list contains a cycle? How do you find the starting node of the cycle? (solution)**

- **How do you reverse a linked list? (solution)**
- **How do you reverse a singly linked list without recursion? (solution)**
- **How are duplicate nodes removed in an unsorted linked list? (solution)**
- **How do you find the length of a singly linked list? (solution)**
- **How do you find the third node from the end in a singly linked list? (solution)**
- **How do you find the sum of two linked lists using Stack? (solution)**

(String)

- **How do you print duplicate characters from a string? (solution)**
- **How do you check if two strings are anagrams of each other? (solution)**
- **How do you print the first non-repeated character from a string? (solution)**
- **How can a given string be reversed using recursion? (solution)**
- **How do you check if a string contains only digits? (solution)**
- **How are duplicate characters found in a string? (solution)**
- **How do you count a number of vowels and consonants in a given string? (solution)**
- **How do you count the occurrence of a given character in a string? (solution)**
- **How do you find all permutations of a string? (solution)**
- **How do you reverse words in a given sentence without using any library method? (solution)**
- **How do you check if two strings are a rotation of each other? (solution)**
- **How do you check if a given string is a palindrome? (solution)**

(Binary Tree)

- **How is a binary search tree implemented? (solution)**

- **How do you perform preorder traversal in a given binary tree? (solution)**
- **How do you traverse a given binary tree in preorder without recursion? (solution)**
- **How do you perform an inorder traversal in a given binary tree? (solution)**
- **How do you print all nodes of a given binary tree using inorder traversal without recursion? (solution)**
- **How do you implement a postorder traversal algorithm? (solution)**
- **How do you traverse a binary tree in postorder traversal without recursion? (solution)**
- **How are all leaves of a binary search tree printed? (solution)**
- **How do you count a number of leaf nodes in a given binary tree? (solution)**
- **How do you perform a binary search in a given array? (solution)**

Other DS Questions

- **How is a bubble sort algorithm implemented? (solution)**
- **How is an iterative quicksort algorithm implemented? (solution)**
- **How do you implement an insertion sort algorithm? (solution)**
- **How is a merge sort algorithm implemented? (solution)**
- **How do you implement a bucket sort algorithm? (solution)**
- **How do you implement a counting sort algorithm? (solution)**
- **How is a radix sort algorithm implemented? (solution)**
- **How do you swap two numbers without using the third variable? (solution)**
- **How do you check if two rectangles overlap with each other? (solution)**
- **How do you design a vending machine? (solution)**

****

Database Questions

Q #1) What do you understand by 'Database'?

Ans: Database is an organized collection of related data where the data is stored and organized to serve some specific purpose.

For ***Example***, A librarian maintains a database of all the information related to the books that are available in the library.

Q #2) Define DBMS.

Ans: DBMS stands for Database Management system. It is a collection of application programs which allow the user to organize, restore and retrieve information about data efficiently and as effectively as possible.

Some of the popular DBMS's are MySql, Oracle, Sybase, etc.

Q #3) Define RDBMS.

Ans: Relational Database Management System(RDBMS) is based on a relational model of data that is stored in databases in separate tables and they are related to the use of a common column. Data can be accessed easily from the relational database using Structured Query Language (SQL).

Q #4) Enlist the advantages of DBMS.

Ans: The Advantages of DBMS includes:

- Data is stored in a structured way and hence redundancy is controlled.
- Validates the data entered and provide restrictions on unauthorized access to the database.
- Provides backup and recovery of the data when required.
- Provides multiple user interfaces.

Q #5) What do you understand by Data Redundancy?

Ans: Duplication of data in the database is known as Data redundancy. As a result of Data Redundancy, duplicated data is present at various locations, hence it leads to wastage of the storage space and the integrity of the database is destroyed.

Q #6) What are the various types of relationships in Database? Define them.

Ans: There are 3 types of relationships in Database:

- **One-to-one:** One table has the relationship with another table having the similar kind of column. Each primary key relates to only one or no record in the related table.

- **One-to-many:** One table has a relationship with another table that has primary and foreign key relation. The primary key table contains only one record that relates to none, one or many records in the related table.
- **Many-to-many:** Each record in both the tables can relate to many numbers of record in another table.

Q #7) Explain Normalization and De-Normalization.

Ans*:* Normalization is the process of removing the redundant data from the database by splitting the table in a well-defined manner in order to maintain data integrity. This process saves much of the storage space.

De-normalization is the process of adding up redundant data on the table in order to speed up the complex queries and thus achieve better performance.

Q #8) What are the different types of Normalization?

Ans: Different Types of Normalization are:

- **First Normal Form (1NF):** A relation is said to be in 1NF only when all the entities of the table contain unique or atomic values.
- **Second Normal Form (2NF):** A relation is said to be in 2NF only if it is in 1NF and all the non-key attribute of the table is fully dependent on the primary key.
- ***Third Normal Form (3NF):*** A relation is said to be in 3NF only if it is in 2NF and every non-key attribute of the table is not transitively dependent on the primary key.

Q #9) What is BCNF?

Ans: BCNF is Boyce Code Normal form. It is the higher version of 3Nf which does not have any multiple overlapping candidate keys.

Q #10) What is SQL?

Ans: Structured Query language, SQL is an ANSI(American National Standard Institute) standard programming language which is designed specifically for storing and managing the data in the relational database management system (RDBMS) using all kinds of data operations.

Q #11) How many SQL statements are used? Define them.

Ans: SQL statements are basically divided into three categories, DDL, DML, and DCL.

They can be defined as:

Data Definition Language (DDL) commands are used to define the structure that holds the data. These commands are auto-committed i.e. changes done by the DDL commands on the database are saved permanently.

Data Manipulation Language (DML) commands are used to manipulate the data of the database. These commands are not auto-committed and can be rolled back.

Data Control Language (DCL) commands are used to control the visibility of the data in the database like revoke access permission for using data in the database.

Q #12) Enlist some commands of DDL, DML, and DCL.

Ans: Data Definition Language (DDL) commands:

- CREATE to create a new table or database.
- ALTER for alteration.
- Truncate to delete data from the table.
- DROP to drop a table.
- RENAME to rename a table.

Data Manipulation Language (DML) commands:

- INSERT to insert a new row.
- UPDATE to update an existing row.
- DELETE to delete a row.
- MERGE for merging two rows or two tables.

Data Control Language (DCL) commands:

- COMMIT to permanently save.
- ROLLBACK to undo the change.
- SAVEPOINT to save temporarily.

Q #13) Define DML Compiler.

Ans: DML compiler translates DML statements in a query language into a low-level instruction and the generated instruction can be understood by Query Evaluation Engine.

Q #14) What is DDL interpreter?

Ans: DDL Interpreter interprets the DDL statements and records the generated statements in the table containing metadata.

Q #15) Enlist the advantages of SQL.

Ans: Advantages of SQL are:

- Simple SQL queries can be used to retrieve a large amount of data from the database very quickly and efficiently.
- SQL is easy to learn and almost every DBMS supports SQL.
- It is easier to manage the database using SQL as no large amount of coding is required.

Q #16) Explain the terms 'Record', 'Field' and 'Table' in terms of database.

Ans: Record: Record is a collection of values or fields of a specific entity. Eg. An employee, Salary account, etc.

Field: A field refers to an area within a record which is reserved for a specific piece of data. Eg. Employee ID.

Table: Table is the collection of records of specific types. E.g. Employee table is a collection of record related to all the employees.

Q #17) What do you understand by Data Independence? What are its two types?

Ans: Data Independence refers to the ability to modify the schema definition in one level in such a way that it does not affect the schema definition in the next higher level.

The 2 types of Data Independence are:

- **Physical Data Independence**: It modifies the schema at the physical level without affecting the schema at the conceptual level.
- **Logical Data Independence:** It modifies the schema at the conceptual level without affecting or causing changes in the schema at the view level.

Q #18) Define the relationship between 'View' and 'Data Independence'.

Ans: View is a virtual table which does not have its data on its own rather the data is defined from one or more underlying base tables.

Views account for logical data independence as the growth and restructuring of base tables is not reflected in views.

Q #19) What are the advantages and disadvantages of views in the database?

Ans: Advantages of Views:

- As there is no physical location where the data in views is stored, it generates output without wasting resources.
- Data access is restricted as it does not allow commands like insertion, updation, and deletion.

Disadvantages of Views:

- View becomes irrelevant if we drop a table related to that view.
- More memory is occupied when the view is created for large tables.

Q #20) What do you understand by Functional dependency?

Ans: A relation is said to be in Functional dependency when one attribute uniquely defines another attribute.

For Example, R is a Relation, X and Y are two attributes. T1 and T2 are two tuples. Then,

T1[X]=T2[X] and T1[Y]=T2[Y] means the value of component X uniquely define the value of component Y.

Also, X->Y means Y is functionally dependent on X.

Q #21) When is functional dependency said to be the fully functional dependency?

Ans: To fulfill the criteria of fully functional dependency, the relation must meet the requirement of functional dependency.

A functional dependency ‘A’ and ‘B’ is said to be fully functional dependent when removal of any attribute say ‘X’ from ‘A’ means the dependency does not hold anymore.

Q #22) What do you understand by E-R model?

Ans: E-R model is an Entity-Relationship model which defines the conceptual view of the database.

E-R model basically shows the real world entities and their association/relations. Entities here represent the set of attributes in the database.

Q #23) Define Entity, Entity type, and Entity set.

Ans: Entity can be anything, be it a place, class or object which has an independent existence in the real world.

Entity type represents a set of entities which have similar attributes.

Entity set in the database represents a collection of entities having a particular entity type.

Q #24) Define Weak Entity set.

Ans: Weak entity set is the one whose primary key comprises of its partial key as well as the primary key of its parent entity.

This is the case because the entity set may not have sufficient attributes to form a primary key.

Q #25) Explain the terms ‘Attribute’ and ‘Relations’

Ans: Attribute describes the properties or characteristics of an entity. For ***Example***, Employee ID, Employee Name, Age, etc., can be attributes of the entity Employee.

Relation is a two-dimensional table containing a number of rows and columns where every row represents a record of the relation. Here, rows are also known as ‘Tuples’ and columns are known as ‘Attributes’.

Q #26) What are VDL and SDL?

Ans: VDL is View Definition language which represents user views and their mapping to the conceptual schema.

SDL is Storage Definition Language which specifies the mapping between two schemas.

Q #27) Define Cursor and its types.

Ans: Cursor is a temporary work area which stores the data as well as the result set occurred after manipulation of data retrieved. A cursor can hold only one row at a time.

The 2 types of Cursor are:

Implicit cursors are declared automatically when DML statements like INSERT, UPDATE, DELETE is executed.

Explicit cursors have to be declared when SELECT statements which are returning more than one row are executed.

Q #28) What is Database transaction?

Ans: Sequence of operation performed which changes the consistent state of the database to another is known as the database transaction. After the completion of the transaction, either the successful completion is reflected in the system or the transaction fails and no change is reflected.

Q #29) Define Database Lock and its types.

Ans: Database lock basically signifies the transaction about the current status of the data item i.e. whether that data is being used by other transactions or not at the present point of time.

There are two types of Database lock which are **Shared Lock and Exclusive Lock.**

Q #30) What is Data Warehousing?

Ans: The storage as well as access to data, that is being derived from the transactions and other sources, from a central location in order to perform the analysis is called Data Warehousing.

Q #31) What do you understand by Join?

Ans: Join is the process of explaining the relationship between different tables by combining columns from one or more table having common values in each. When a table joins with itself, it is known as Self Join.

Q #32) What do you understand by Index hunting?

Ans: Index hunting is the process of boosting the collection of indexes which help in improving the query performance as well as the speed of the database.

Q #33) How to improve query performance using Index hunting?

Ans: Index hunting help in improving query performance by:

- Using query optimizer to coordinate queries with the workload.
- Observing the performance and effect of index and query distribution.

Q #34) Differentiate between 'Cluster' and 'Non-cluster' index.

Ans: Clustered Index alters the table and reorders the way in which the records are stored in the table. Data retrieval is made faster by using the clustered index.

A Non-clustered index does alter the records that are stored in the table but creates a completely different object within the table.

Q #35) What are the disadvantages of a Query?

Ans: Disadvantages of a Query are:

- Indexes are not present.
- Stored procedures are excessively compiled.
- Difficulty in interfacing.

Q #36) What do you understand by Fragmentation?

Ans: Fragmentation is a feature which controls the logical data units, also known as fragments that are stored at different sites of a distributed database system.

Q #37) Define Join types.

Ans: Given below are the types of Join, which are explained with respect to the tables as an ***Example***:

employee table:

EmpID	EmpName
1000	Rohan
1001	Shruti
1002	Nikhil
1003	Naveen

employee_info table:

EmpID	Address
1000	Delhi
1001	Mumbai
1002	Delhi
1003	Kolkata

1) Inner JOIN: Inner JOIN is also known as a simple JOIN. This SQL query returns result from both the tables having a common value in rows.

SQL Query:

SELECT * from employee, employee_info WHERE employee.EmpID = employee_info.EmpID ;

Result:

EmpID	EmpName	EmpID	Address
1000	Rohan	1000	Delhi
1001	Shruti	1000	Delhi
1002	Nikhil	1000	Delhi
1003	Naveen	1000	Delhi
1000	Rohan	1001	Mumbai
1001	Shruti	1001	Mumbai
1002	Nikhil	1001	Mumbai
1003	Naveen	1001	Mumbai
1000	Rohan	1002	Delhi
1001	Shruti	1002	Delhi
1002	Nikhil	1002	Delhi
1003	Naveen	1002	Delhi
1000	Rohan	1003	Kolkata
1001	Shruti	1003	Kolkata
1002	Nikhil	1003	Kolkata
1003	Naveen	1003	Kolkata

2) Natural JOIN: This is a type of Inner JOIN which returns results from both the tables having same data values in the columns of both the tables to be joined.

SQL Query:

SELECT * from employee NATURAL JOIN employee_info;

Result:

EmpID	EmpName	Address
1000	Rohan	Delhi
1001	Shruti	Mumbai
1002	Nikhil	Delhi
1003	Naveen	Kolkata

3) Cross JOIN: Cross JOIN return results as all the records where each row from the first table is combined with each row of the second table.

SQL Query:

SELECT * from employee CROSS JOIN employee_info;

Result:

Let us do some modification in the above tables to understand Right JOIN, Left JOIN, and Full JOIN.

employee table:

EmpID	EmpName
1000	Rohan
1001	Shruti
1002	Nikhil
1003	Naveen
1004	Shikha
1005	Shalu

employee_info table:

EmpID	Address
1000	Delhi
1001	Mumbai
1002	Gurgaon
1003	Kolkata
1006	Noida
1007	Kerala

1) Right JOIN: Right JOIN is also known as Right Outer JOIN. This returns all the rows as a result from the right table even if the JOIN condition does not match any records in the left table.

SQL Query:

SELECT * from employee RIGHT OUTER JOIN employee_info on (employee.EmpID = employee_info.EmpID);

Result:

EmpID	EmpName	EmpID	Address
1000	Rohan	1000	Delhi
1001	Shruti	1001	Mumbai
1002	Nikhil	1002	Delhi
1003	Naveen	1003	Kolkata
Null	Null	1006	Noida
Null	Null	1007	Kerala

2) Left JOIN: Left JOIN is also known as Left Outer JOIN. This returns all the rows as a result of the left table even if JOIN condition does not match any records in the right table. This is exactly the opposite of Right JOIN.

SQL Query:

SELECT * from employee LEFT OUTER JOIN employee_info on (employee.EmpID = employee_info.EmpID);

Result:

EmpID	EmpName	EmpID	Address
1000	Rohan	1000	Delhi
1001	Shruti	1001	Mumbai
1002	Nikhil	1002	Delhi
1003	Naveen	1003	Kolkata
1004	Shikha	null	Null
1005	Shalu	null	Null

3) Outer/Full JOIN: Full JOIN return results in combining the result of both the Left JOIN and Right JOIN.

SQL Query:

SELECT * from employee FULL OUTER JOIN employee_info on (employee.EmpID = employee_info.EmpID);

Result:

EmpID	EmpName	EmpID	Address
1000	Rohan	1000	Delhi
1001	Shruti	1001	Mumbai
1002	Nikhil	1002	Delhi
1003	Naveen	1003	Kolkata
1004	Shikha	null	Null
1005	Shalu	null	Null
Null	Null	1006	Noida
Null	Null	1007	Kerala

Q #38) What do you understand by 'Atomicity' and 'Aggregation'?

Ans: Atomicity is the condition where either all the actions of the transaction are performed or none. This means, when there is an incomplete transaction, database management system itself will undo the effects done by the incomplete transaction.

Aggregation is the concept of expressing the relationship with the collection of entities and their relationships.

Q #39) Define Phantom deadlock.

Ans: Phantom deadlock detection is the condition where the deadlock does not actually exist but due to a delay in propagating local information, deadlock detection algorithms identify the deadlocks.

Q #40) Define checkpoint.

Ans: Checkpoint declares a point before which all the logs are stored permanently in the storage disk and is the inconsistent state. In the case of crashes, the amount of work and time is saved as the system can restart from the checkpoint.

Q #41) What is Database partitioning?

Ans: Database partitioning is the process of partitioning tables, indexes into smaller pieces in order to manage and access the data at a finer level.

This process of partitioning reduces the cost of storing a large amount of data as well as enhances the performance and manageability.

Q #42) Explain the importance of Database partitioning.

Ans: The importance of Database partitioning are:

- Improves query performance and manageability.
- Simplifies common administration tasks.
- Acts as a key tool for building systems with extremely high availability requirements.
- Allows accessing a large part of a single partition.

Q #43) Explain Data Dictionary.

Ans: Data dictionary is a set of information describing the content and structure of the tables and database objects. The job of the information stored in the data dictionary is to control, manipulate and access the relationship between database elements.

Q #44) Explain Primary Key and Composite Key.

Ans: Primary key is that column of the table whose every row data is uniquely identified. Every row in the table must have a primary key and no two rows can have the same primary key. Primary key value can never be null nor can be modified or updated.

Composite Key is a form of the candidate key where a set of columns will uniquely identify every row in the table.

Q #45) What do you understand by Unique key?

Ans: A Unique key is same as the primary key whose every row data is uniquely identified with a difference of null value i.e. Unique key allows one value as NULL value.

Q #46) What do you understand by Database Triggers?

Ans: A set of commands that automatically get executed when an event like Before Insert, After Insert, On Update, On Delete of row occurs in a table is called as Database trigger.

Q #47) Define Stored procedures.

Ans: A Stored procedure is a collection of pre-compiled SQL Queries, which when executed denotes a program taking input, process and gives the output.

Q #48) What do you understand by B-Trees?

Ans: B-Tree represents the data structure in the form of a tree for external memory that reads and writes large blocks of data. It is commonly used in databases and file systems where all the insertions, deletions, sorting, etc., are done in logarithmic time.

Q #49) Name the different data models that are available for database systems.

Ans: Different data models are:

- Relational model
- Network model
- Hierarchical model

Q #50) Differentiate between 'DELETE', 'TRUNCATE' and 'DROP' commands.

Ans: After the execution of **'DELETE'** operation, COMMIT and ROLLBACK statements can be performed to retrieve the lost data.

After the execution of **'TRUNCATE'** operation, COMMIT, and ROLLBACK statements cannot be performed to retrieve the lost data.

'DROP' command is used to drop the table or key like the primary key/foreign key.

Q #51) Based on the given table, solve the following queries.

Employee table

empId	empName	Age	Address
1001	Rohan	26	Delhi
1002	Ankit	30	Gurgaon
1003	Gaurav	27	Mumbai
1004	Raja	32	Nagpur

1) Write the SELECT command to display the details of the employee with empid as 1004.

Ans:

SELECT empId, empName, Age, Address from Employee WHERE empId = 1004;

Result:

empId	empName	Age	Address
1004	Raja	32	Nagpur

2) Write the SELECT command to display all the records of table Employee.

Ans:

SELECT * from Employee;

Result:

empId	empName	Age	Address
1001	Rohan	26	Delhi
1002	Ankit	30	Gurgaon
1003	Gaurav	27	Mumbai
1004	Raja	32	Nagpur

3) Write the SELECT command to display all the records of the employee whose name starts with the character 'R'.

Ans:

SELECT * from Employee WHERE empName LIKE 'R%';

Result:

empId	empName	Age	Address
1001	Rohan	26	Delhi
1004	Raja	32	Nagpur

4) Write a SELECT command to display id, age and name of the employees with their age in both ascending and descending order.

Ans:

SELECT empId, empName, Age from Employee ORDER BY Age;

Result:

empId	empName	Age
1001	Rohan	26
1003	Gaurav	27
1002	Ankit	30
1004	Raja	32

SELECT empId, empName, Age from Employee ORDER BY Age Desc;

Result:

empId	empName	Age
1004	Raja	32
1002	Ankit	30
1003	Gaurav	27
1001	Rohan	26

5) Write the SELECT command to calculate the total amount of salary on each employee from the below Emp table.

Emp table:

empId	empName	Age	Salary
1004	Raja	32	6000
1002	Ankit	30	8000
1003	Gaurav	27	5000
1001	Rohan	26	12000
1005	Sneha	29	9000

Ans:

SELECT empName, SUM(Salary) from Emp GROUP BY empName;

Result:

empName	Salary
Ankit	8000
Gaurav	5000
Raja	6000
Rohan	12000
Sneha	9000

****

Java :

****

(Note : Question may be repeat in one or another way. And Some may have answers also.)

- Name some of the characteristics of OO programming languages
- What are the access modifiers you know? What does each one do?

- What is the difference between overriding and overloading a method in Java?
- What's the difference between an `Interface` and an `abstract` class?
- Can an `Interface` extend another `Interface`?
- What does the `static` word mean in Java?
- Can a static method be overridden in Java?
- What is Polymorphism? What about Inheritance?
- Can a constructor be inherited?
- Do objects get passed by reference or value in Java? Elaborate on that.
- What's the difference between using == and `.equals` on a string?
- What is the `hashCode()` and `equals()` used for?
- What does the interface Serializable do? What about Parcelable in Android?
- Why are `Array` and `ArrayList` different? When would you use each?
- What's the difference between an `Integer` and `int`?
- What is a `ThreadPool`? Is it better than using several "simple" threads?
- What the difference between local, instance and class variables?
- What is reflection?
- What is dependency injection? Can you name a few libraries? (Have you used any?)
- What are strong, soft and weak references in Java?
- What does the keyword `synchronized` mean?
- Can you have "memory leaks" on Java?
- Do you need to set references to null on Java/Android?
- What does it means to say that a `String` is immutable?
- What are `transient` and `volatile` modifiers?
- What is the `finalize()` method?
- How does the `try{} finally{}` works?
- What is the difference between `instantiation` and `initialisation` of an object?
- When is a static block run?

- Why are Generics are used in Java?
- Can you mention the design patterns you know? Which of those do you normally use?
- Can you mention some types of testing you know?
- How does `Integer.parseInt()` works?
- Do you know what is the “double check locking” problem?
- Do you know the difference between `StringBuffer` and `StringBuilder`?
- How is a `StringBuilder` implemented to avoid the immutable string allocation problem?
- What does `Class.forName` method do?
- What is Autoboxing and Unboxing?
- What’s the difference between an `Enumeration` and an `Iterator`?
- What is the difference between fail-fast and fail safe in Java?
- What is PermGen in Java?
- What is a Java priority queue?
- Is performance influenced by using the same number in different types: `Int`, `Double` and `Float`?
- What is the Java Heap?
- What is daemon thread?
- Can a dead thread be restarted?

Preparing for a Java Interview

Java interviews can vary depending on the candidate’s experience. For example, junior Java developers with one-to-four years of experience may experience questions on topics like fundamentals, API, data structure, and algorithms.

Senior Java developers with more than five or six years of experience will be asked more questions about concurrent programming, Java concurrency API, JVM internals, GC tuning and Java performance.

Multithreading, Concurrency, and Thread Questions and Answers

1. **Is it possible to make array volatile in Java?**

Answer: Yes, it is possible to make an array volatile in Java, but only the reference which is pointing to an array, not the whole array. Therefore, if one thread changes the reference variable points to another array, which will provide a volatile guarantee.

However, if several threads are altering particular array elements, there won't be any happens before assurance provided by the volatile modifier for such modification.

If the purpose is to provide memory visibility guarantee for individual indices of the array, volatile is of no practical use for you. Instead, you must rely on an alternative mechanism for thread-safety in that particular case, e.g. use of a synchronised keyword.

2. **From the two, which would be easier to write: synchronisation code for ten threads or two threads?**

Answer: Both will have the same level of complexity regarding writing the code because synchronisation is independent of the number of threads, although the choice of synchronisation could be subject to the number of threads because this presents more conflict.

Therefore, you would opt for an advanced synchronisation technique, e.g. lock stripping, which requires more intricate code and proficiency.

3. **How would you call wait() method? Would you use if block or loop, and why?**

Answer: wait() method should always be called in loop. It is likely that, until the thread gets CPU to start running again, the condition may not hold. Therefore, it is always advised to check the condition in loop before continuing.

4. **What is defined as false sharing in the context of multithreading?**

Answer: False sharing is known to be one of the familiar performance issues on multi-core systems, whereby each process has a local cache.

False sharing can be hard to identify since the thread may be retrieving completely different global variables that occur to be fairly close together in memory.

Similar to many other concurrency issues, the main way to avoid false sharing is to carefully review your code and supporting your data structure with the size of a cache line.

5. **What is busy spin, and why should you use it?**

Answer: Busy spin is known as one of the techniques to wait for events without freeing CPU. This is often done to avoid losing data in CPU cache, which could get lost if the thread is paused and resumed in some other core.

As a result, if you are working on a low latency system where your order processing thread isn't in any particular order, rather than sleeping or calling wait(), you can just loop and then review the queue for new messages.

This is only valuable if you need to wait for a short amount of time, e.g. in microseconds or nanoseconds. LMAX Disruptor frameworks, a high-performance inter-thread messaging library has a BusySpinWait Strategy, which is centred on this model and uses a busy spin loop for EventProcessors waiting on the barrier.

LMAX Disruptor frameworks, a high-performance inter-thread messaging library has a BusySpinWait Strategy, which is centred on this model and uses a busy spin loop for EventProcessors waiting on the barrier.

6. **How do you take thread dump in Java?**

Answer: By using **kill -3 PID** in Linux, where PID is the process id of Java process, you can take a thread dump of Java application. In Windows, you can press **Ctrl + Break.**

This will instruct JVM to print thread dump in standard out, and it may go to console or log file depending on your application configuration.

7. **Is Swing thread-safe?**

Answer: No, Swing is not thread-safe. You aren't able to update Swing components, e.g. JTable, JList or Jpanel from any thread. In fact, they must be updated from a GUI or AWT thread. This is why Swing's provide

In fact, they must be updated from a GUI or AWT thread. This is why Swing's provide invokeAndWait() and invokeLater() method to request GUI update from alternative threads.

These methods put update requests in AWT threads queue and wait for the update or return straight away for an asynchronous update.

8. **Describe what a thread-local variable is in Java**

Answer: Thread-local variables are variables restricted to a thread. It is like thread's own copy which is not shared between a multitude of threads. Java offers a ThreadLocal class to upkeep thread-local variables. This is one of the many ways to guarantee thread-safety.

However, it is important to be mindful while using a thread local variable in a controlled environment, e.g. with web servers where worker thread outlives any application variable.

Any thread local variable which is not taken away once its work is done can hypothetically cause a memory leak in Java application.

9. **What is the difference between sleep and wait in Java?**

Answer: Both are used to pause thread that is currently running, however, sleep() is meant for short pause because it does not release lock, while wait() is meant for conditional wait.

This is why it releases lock, which can then be developed by a different thread to alter the condition of which it is waiting.

10. **What is defined as an immutable object? How would you create an immutable object in Java?**

Answer: Immutable objects are defined as those whose state cannot be changed once it has been made, any alteration will result in a new object, e.g. String, Integer, and other wrapper class.

11. **What's the difference between Callable and Runnable?**

Answer: Both of these are interfaces used to carry out task to be executed by a thread. The main difference between the two interfaces is that Callable can return a value, while Runnable cannot. Another difference is that Callable can throw a checked exception, while Runnable cannot. Runnable has been around since Java 1.0, while Callable was introduced as part of Java 1.5.

12. **What do the Thread.class methods run() and start() do?**

Answer: Thread.run() will execute in the calling thread, i.e. a new thread is not created, the code is executed synchronously. Thread.start() will execute the same code, but in a new asynchronous thread.

13. **What is the right data type to represent a price in Java?**

Answer: If memory is not a concern and performance is not critical, BigDecimal will be the right data type represent a price in Java. If not, double with predefined precision.

14. **How would you convert bytes to String?**

Answer: To convert bytes to String, you would use Strong constructor which accepts byte[].

However, you should be mindful of the right character encoding otherwise the platform's default character encoding will be used, which may not necessarily be the same.

15. **Is it possible to cast an int value into a byte variable? What would happen if the value of int is larger than byte?**

Answer: Yes, it is possible but int is 32 bit long in Java, while byte is 8 bit long in Java. Therefore when you can cast an int to byte higher, 24 bits are gone and a byte can only hold a value between -128 to 128.

16. **Which class contains method: Cloneable or Object?**

Answer: java.lang.Cloneable is marker interface and does not contain at all any method. Clone method is well-defined in the object class.

Remember that clone() is a native method, therefore it is applied in C or C++ or any other native programming language.

17. **Is ++ operator thread-safe in Java?**

Answer: ++ is not thread-safe in Java because it involves multiple commands such as reading a value, implicating it, and then storing it back into memory.

This can be overlapped between multiple threads.

18. **Is it possible to store a double value in a long variable without casting?**

Answer: No, it is not possible to store a double value into a long variable without casting since the range of double is more, meaning you would need to type cast.

19. **Which one will take more memory: an int or Integer?**

Answer: An integer object will take more memory as it stores metadata overhead about the object. An int is primitive, therefore it takes less space.

20. **Why is String immutable in Java?**

Answer: String is immutable in Java since Java designer thought that String will be greatly used, making it immutable. It lets some optimisation easy sharing, and same String object between multiple clients.

A key step in that direction was the idea of putting away String literals in String pool. The aim was to moderate temporary String object by sharing them and in order to share, they must have to be from immutable class.

It is worth noting, that it isn't possible to share a mutable project with two parties which are unfamiliar to each other.

21. **Is it possible to use String in the switch case?**

Answer: Yes, this is possible from Java 7 onward. String can be used in switch case, but it is just syntactic sugar. Internal string hashcode is used for the switch.

22. **What is constructor chaining in Java?**

Answer: Constructor chaining in Java is when you call one constructor from another. This generally occurs when you have multiple, overloaded constructor in the class.

23. **What are the default values of Java primitives?**

Answer: byte – 0, int – 0, boolean – false, short – 0, long – 0L, float – 0.0f, double – 0.0d, char – '\u0000'

24. **How is the transient keyword used in Java?**

Answer: The transient keyword is used to indicate that a field in class should not be serialized (used with the Serializable interface)

25. **What is the size of int in 64-bit JVM?**

Answer: The size of an int variable is constant in Java, it is always 32-bit regardless of platform. This means the size of primitive int is identical in both 32-bit and 64-bit Java Virtual Machine.

26. **What is the size of an int variable in 32-bit and 64-bit JVM?**

Answer: The size of int is identical in both 32-bit and 64-bit JVM, and it is always 32-bits or 4 bytes.

27. **How does WeakHashMap work?**

Answer: WeakHashMap operates like a normal HashMap but uses WeakReference for keys. Meaning if the key object does not devise any reference then both key/value mapping will become appropriate for garbage collection.

28. **How do you identify if JVM is 32-bit or 64-bit from Java Program?**

Answer: This can be identified by checking some system properties such as sun.arch.data.model or os.arch.

29. **What is the maximum heap size of 32-bit and 64-bit JVM?**

Answer: In theory, the maximum heap memory you can assign to a 32-bit JVM is 2^32, which is 4GB, but practically the bounds are much smaller.

It also depends on operating systems, e.g. from 1.5GB in Windows to almost 3GB in Solaris. 64-bit JVM allows you to stipulate larger heap size, hypothetically 2^64, which is quite large but practically you can specify heap space up to 100GBs.

30. **Explain the difference between JRE, JDK, JVM, and JIT.**

Answer: JRE is an abbreviation of Java Runtime Environment that consist of sets of files needed by JVM throughout the runtime.

JVM is an abbreviation of Java Virtual Machine which delivers the runtime environment for collected Java Bytecode. JVM is in control of the conversion of the bytecode into machine-readable code.

JDK is an abbreviation for Java Development Kit which contains JRE including development tools for the purpose of development. JDK is required to write and execute a Java code.

JIT is an abbreviation of Just in Time compilation, and this helps to improve the performance of Java application by converting Java bytecode into native code when they cross certain threshold, i.e. the mostly hot code is transformed into native code.

31. **Explain Java Heap Space and Garbage collection.**

Answer: When a Java process has started using Java command, memory is distributed to it. Part of this memory is used to build heap space, which is used to assign memory to objects every time they are formed in the program.

Part of this memory is used to build heap space, which is used to assign memory to objects every time they are formed in the program.

Garbage collection is the procedure inside JVM which reclaims memory from dead objects for future distribution.

32. **Can you guarantee the garbage collection process?**

Answer: No, the garbage collection cannot be guaranteed, though you can make a request using System.gc() or Runtime.gc() method.

33. **How do you locate memory usage from a Java program? How much of the percent is used?**

Answer: You can use memory related methods from java.lang.Runtime class to get the free memory, total memory and maximum heap memory in Java.

From using these methods, you can find out how much percent of the heap is used and how much heap space is outstanding.

Runtime.freeMemory() return the amount of free memory in bytes, Runtime.totalMemory() returns the total memory in bytes and Runtime.maxMemory() returns the maximum memory in bytes.

34. **What is the difference between Stack and Heap in Java?**

Answer: Stack and Heap are different memory areas in the JVM, and they are used for different purposes.

The stack is usually much smaller than heap memory and also isn't shared amongst multiple threads, but heap is shared among all threads in JVM.

35. **What is the difference between 'a == b' and 'a.equals(b)'?**

Answer: The 'a = b' does object reference matching if both **a** and **b** are an object and only return true if both are pointing to the same object in the heap space. However, a.equals(b) is used for logical mapping and it is likely from an object to supersede this method to provide logical equality.

For example, String class overrides this equals() method so that you can associate two Strings, which are not the same object but covers the same letters.

36. **What is a.hashCode() used for? How is it related to a.equals(b)?**

Answer: hashCode() method returns an int hash value corresponding to an object. It is used in hash-based collection classes e.g. HashTable, HashMap, LinkedHashMap. It is very closely related to equals() method.

According to the Java specification, two objects which are identical to each other using equals() method needs to have the same hash code.

37. **What is the difference between final, finalize and finally?**

Answer: Final is a modifier which you can apply to variable, methods, and classes. If you create a variable final, this means its value cannot be changed once initialised.

Finalise is a method, which is called just before an object is a garbage collected, allowing it a final chance to save itself, but the call to finalise is not definite.

Finally is a keyword which is used in exception handling, along with try and catch. The finally block is always implemented regardless of whether an exception is thrown from try block or not.

38. **What is a compile time constant in Java? What is the risk of using it?**

Answer: Public static final variables are also known as the compile time constant, the public is optional there. They are substituted with actual values at compile time because compiler recognises their value up-front, and also recognise that it cannot be altered during runtime.

One of the issues is that if you choose to use a public static final variable from in-house or a third party library, and their value changed later, then your client will still be using the old value even after you deploy a new version of JARs.

This can be avoided by ensuring you compile your program when you upgrade dependency JAR files.

39. **What happens when a finally block has a return statement?**

Answer: The returned value will override any value returned by the corresponding try block.

40. **Can you override a static method?**

Answer: No, static methods are not overridable.

41. **What is the difference between List, Set, Map and Queue in Java?**

Answer: List, Set, and Map are three significant interfaces of Java collection framework.

Set provides an unordered collection of unique objects i.e. set does not allow duplicates, while Map provides a data structure based on key-value pair and hashing.

The difference between List and Set interface in Java is that List allows duplicates while Set does not allow duplicates. All implementation of Set honour this agreement. Map holds two objects per entry e.g. key and value, and it may contain duplicate values but keys are always unique.

One more difference between List and Set is that List is an ordered collection, List's contract maintains insertion order or element. Set is an unordered collection, therefore you get no assurance on which order element will be stored.

Nevertheless, some of the set implementation (e.g. LinkedHashSet) retains order.

A queue is also ordered, but you will only ever touch elements at one end. All elements get inserted at the 'end' and removed from the 'beginning' (or head) of the queue.

You are able to find out how many elements are in the queue, but you are not able to find out what, for example, the 'third' element is.

42. **What is the difference between poll() and remove() method?**

Answer: Both poll() and remove() take out the object from the Queue but if poll() fails, then it returns null. However, if remove() fails, it throws exception.

43. **What is the difference between LinkedHashMap and PriorityQueue in Java?**

Answer: PriorityQueue guarantees that the lowest or highest priority element always remains at the head of the queue. However, LinkedHashMap maintains the order on which elements are inserted.

When you repeat over a PriorityQueue, iterator does not promise any order but iterator of LinkedHashMap does promise the order on which elements are put in.

44. **What is the difference between ArrayList and LinkedList in Java?**

Answer: The main difference between them is that ArrayList is supported by array data structure, supports random access. LinkedList is backed by linked list data structure and doesn't support random access.

45. **How do you print Array in Java?**

Answer: You can print an array by using the Arrays.toString() and Arrays.deepToString() method. Since Array does not implement to String() by itself, just passing an array to System.out.printIn() will not print its content but Array.to.String will print each element.

46. **Is LinkedList in Java a doubly or singly linked list?**

Answer: LinkedList is a doubly linked list, and you can review the code in JDK. In Eclipse, you are able to use the shortcut, **Ctrl + T** to directly open this class in editor.

47. **What is the difference between Hashtable and HashMap?**

Answer: There are several differences between the two classes, including:

- Hashtable is a legacy class and current from JDK 1, HashMap was introduced and added later.
- Hashtable is synchronised and slower whereas HashMap is not synchronised and faster.
- Hashtable does not allow null keys but HashMap allows one null key.

48. **How does HashSet work internally in Java?**

Answer: HashSet is internally implemented using a HashMap. Since a Map needs a key and value, a default value is used for all keys. Like HashMap, HashSet does not allow identical keys and only one null key – you are only able to store one null object in HashSet.

49. **Is it possible for two unequal objects to have the same hashcode?**

Answer: Yes, two unequal objects can have the same hashcode. This is why collision can occur in hashmap. The equal hashcode contract only says that two equal objects must have the identical hashcode, but there is no indication to say anything about the unequal object.

50. **What is the difference between Comparator and Comparable in Java?**

Answer: The comparable interface is used to define the natural order of object while Comparator is used to describe custom order. Comparable can always be one, but it is possible to have multiple comparators to define a custom order for objects.

From a Java interview point of view, IO is very important. Ideally, you should have a good knowledge of old Java IO, NIO, and NIO2. Additionally, it is worth having knowledge of some operating systems and disk IO fundamentals. The following are some frequently asked questions regarding Java IO:

51. **What is the byte order of ByteBuffer?**

Answer: The byte order is used when reading or writing multibyte values, and when creating buffers that are views of this byte buffer. The order of a new byte buffer is always BIG_ENDIAN.

52. **What is the difference between direct buffer and non-direct buffer in Java?**

Answer: Byte buffer is one of the important class of Java NIO API. This was initially introduced in java.nio package on JDK 1.4. It lets you function on heap byte arrays as well as with direct memory, which occurs outside the JVM.

It lets you function on heap byte arrays as well as with direct memory, which occurs outside the JVM.

The main difference between direct and non-direct byte buffers are their memory location, non-direct byte buffers are just a wrapper around byte array and they reside in Java Heap memory.

Meanwhile, direct byte buffer is outside of JVM and memory is not assigned from the heap.

A byte buffer is either direct or non-direct. Given a direct byte buffer, the Java Virtual Machine will make a best effort to complete native I/O operations directly upon it.

It will try to evade copying the buffer's content to (or from) an intermediate buffer before (or after) each invocation of one of the underlying operating system's native I/O operations.

53. **What is the memory mapped buffer in Java?**

Answer: Java IO has been considerably fast after the introduction of NIO and memory mapped file offers fastest IO operation possible in Java.

A key advantage of memory mapped file is that operating system is responsible for reading and writing and even if your program malfunctioned just after writing into memory. OS will take care of writing content to file.

54. **What is the difference between TCP and UDP protocol?**

Answer: TCP and UDP are two transport layer protocols, which are widely used on the internet for transferring data from one host to another.

TCP is a connection-oriented protocol, whereas UDP is a connectionless protocol. Therefore, a connection is recognised between client and server before they can send data.

With UDP being a connectionless protocol, the point-to-point connection is not recognised before sending messages. For that reason, UDP is more fit for multicast distribution of the message.

There are many differences between TCP and UDP, but during the Java interview it is worth mentioning that TCP is connection oriented, dependable, sluggish, offers definite delivery and preserves the order of messages, while UDP is connectionless, unpredictable, no ordering assurance, but a fast protocol.

TCP is also much higher than UDP, as it transfers more metadata per packer than UDP.

Additionally, the header size of TCP is 20 bytes, compared to 8 bytes header of UDP. If you are in a position where you can't afford to lose any messages, use TCP. If you need high-speed data transmission, where loss of a single packet is acceptable, use UDP.

If you need high-speed data transmission, where loss of a single packet is acceptable, use UDP.

55. **What best practices should you follow while writing multithreaded code in Java?**

Answer: When writing concurrent code in Java, the following are some best practices to be mindful of:

- Always name your thread as this help in debugging.
- Minimise the scope of your of synchronisation. Rather than making the whole method synchronised, be mindful that only the critical section should be synchronised.
- Opt for volatile over synchronisation if you have the option to.
- Use a higher level of concurrency utilities instead of waiting() and notify for inter-thread communication, e.g. BlockingQueue, CountDownLatch and Semaphore.
- Opt for concurrent collection over synchronised collection in Java as this will provide better scalability.

56. **Explain some best practices you would apply while using Collection in Java.**

Answer: When using Collection classes in Java, the following are some best practices to be mindful of:

- Ensure you are using the right collection, e.g. if you need a non-synchronised list, then opt for ArrayList and not Vector.
- Opt for concurrent collection over a synchronised collection because they are more scalable.
- Ensure you are using interface to represent and access a collection e.g. use List to store ArrayList, Map to store HashMap.
- Use iterator to loop over collection.
- Always use generics with collection.

57. **Explain five best practices you would apply while using threads in Java.**

Answer: Although similar to the previous question, be particularly mindful with thread, as you should:

- Always name your thread.
- Prioritise your task and threads by keeping them separate. Use runnable and Callable with thread pool executor.
- Use thread pool.
- Use volatile to indicate compiler about ordering, visibility, and atomicity.
- Avoid thread local variable because improper use of ThreadLocal class in Java can produce a memory leak.

58. **Explain 5 IO best practices.**

Answer: IO is important for overall performance for your Java application, and ideally you should avoid IO in a critical path of your application. The following are Java IO best practices you should be mindful of:

- Use buffered IO classes instead of reading individual bytes and char.
- Use classes from NIO and NIO2.
- Always close streams in final block or use try-with-resource statements.
- Use memory mapped file for faster IO.

59. **Explain a few method overloading best practices in Java.**

Answer: The following are examples of overloading a method in Java to avoid confusion with auto-boxing:

- Don’t overload method where one accepts ints and the other accepts Integer.
- Don’t overload method where a number of arguments are the same and only the order of argument is different.

Use varargs after the overload methods have had more than five arguments.

Date, Time, and Calendar Interview Questions in Java

60. **Is SimpleDateFormat safe to use in the multithreaded program?**

Answer: No, DataFormat and all its implementation including SimpleDateFormat is not thread-safe, hence should not be used in the multithreaded program until external thread-safety measures are applied e.g. confining SimpleDateFormat object into ThreadLocal variable.

If you do not do that, you will get wrong results while analysing or configuring dates in Java.

61. **How would you format a data in Java? i.e. in the DDMMYYY format.**

Answer: This can be done by using either SimpleDateFormat class or java-time library to format a date in Java. DateFormat class lets you format the date on many common formats.

62. **How do you show the time zone in a formatted date In Java?**

Answer: You are able to put time zone information in formatted Date using z attribute of DateFormat class.

63. **What is the difference between java.util.Date and java.sql.Date in Java?**

Answer: Both are known as Date class, but there is some difference between java.util.Date and java.sql.Date, for example, Former is used whenever a Date is required in Java application while later is used to read and store DATE SQL type from the database.

Another significant difference is java.util.Date stores both date and time values, while java.sql.date only stores date information, without any time apart. According to the Javadoc specification, java.sql.date is a thin wrapper around a millisecond value that lets JDBC distinguish as an SQL DATE value.

To fit in with the meaning of SQL DATE, the millisecond values bound by a java.sql.Date instance must be 'normalised'. However, setting the hours, minutes, seconds, and milliseconds to zero in the specific time zone with which the case is connected.

Unit Testing JUnit Interview Questions

64. **How do you test static method?**

Answer: PowerMock library can be used to test static methods in Java.

65. **How do you test a method for an exception using JUnit?**

Answer: One part of unit testing a Java method is checking exception thrown by that method. In a Java unit test, it should really prove correct exception thrown in exceptional case and no exception would be thrown in normal case.

In order to test any Java method for throwing exception in Junit4, you need to make sure that argument provided to that method, from the test must result in expected exception, otherwise JUnit test will fail.

A testing method called speed(), returns speed as distance/time, but before calculating speed it checks whether time and distance are positive or negative and if time is zero or negative it throws IllegalArgumentException.

66. **What is the difference between @Before and @BeforeClass annotation?**

Answer: The code marker @Before is executed before each test, while @BeforeClass runs once before the entire test fixture. If your text class has ten tests, @Before code will be executed ten times, but @BeforeClass will be executed only once.

In general, you would use @BeforeClass when numerous tests are required to share the same computationally expensive setup code. Starting a database connection falls into this category. You are able to move code from @BeforeClass into @Before, but your test run may be delayed.

@BeforeClass is run as static initialiser, therefore it will run before the class instance of your test fixture is created.

67. **What's the difference between unit, integration and functional testing?**

Answer: A unit tests tests a small isolated piece of code such as a method. It doesn't interact with any other systems such as a database or another application.

An integration test tests how your code plays with other systems, such as a database, a cache or a third-party application.

A functional test is the testing of the complete functionality of an application. This may involve the use of an automated tool to carry out more complex user interactions with your system. It tests certain flows/use cases of your application.

68. **What is the difference between state-based unit testing and interaction-based unit testing?**

Answer: State-based unit testing tests that the resulting state of a piece of code under test is as expected. Interaction-based testing tests that the piece of code under tests followed a certain flow or invoked certain methods as expected.

Programming and Coding Questions

69. **How do you check if a String contains only numeric digits?**

Answer: java.lang.String class offers a couple of methods with an inherent support of systematic expression e.g.split method, replaceAll() and matches method. Although this can be used for this purpose, it can have a disadvantage.

They produce a new consistent expression pattern object, each time you call. Since most of the time the pattern can be reused, there is no need to spend time on producing and assembling pattern, which is costly in comparison to testing a String against the pattern.

In terms of reusable patterns, you can receive assistance of java.util.regex package, it offers two class Pattern and Matcher to create pattern and review String alongside that pattern.

To complete this, you need to create a regular expression pattern object. This can be done by passing regular expression String "(.)*(\\d)(.)*" to Pattern.compile() method.

This will then output a compiled version of regular expression String. From using this pattern you can acquire Matcher object to identify if input string passes this systematic expression pattern or not.

70. **How do you write a Java program to convert bytes to long?**

Answer: The byte takes 1 byte of memory and long takes 8 bytes of memory. Assignment 1 byte value to 8 bytes is done indirectly by the JVM.

Byte -> short -> int -> long -> float -> double

The left-side value can be assigned to any right-side value and is done indirectly. The reverse requires explicit casting.

71. **How do you reverse a String in Java without using StringBuffer?**

Answer: To reverse a String in Java, you are able to use rich Java API to quickly reverse contents of any String object. The Java library provides String Buffer and StringBuilder class with reverse() method, which can be used to reverse String in Java.

Since changing between String and StringBuffer is very easy, this is the easiest way presented to reverse String in Java. Reverse is a recursive job, and for that reason you can use recursion as well as loop to reverse String in Java.

72. **How do you check if two given String are anagrams?**

Answer: Anagrams are a mix-up of characters in String e.g. army and mary, stop and pots etc. To identify if Strings are anagram, you will need to get their character array and identify if they are equal or not.

You are able to use indexOf(), substring() and StringBuffer or StringBuilder class to solve this question.

73. **How do you convert String to int in Java?**

Answer: Java provides Integer.parseInt() method to parse a String to an int value. However, there is another way to do this, which takes advantage of the parsing

logic of parseInt() method as well as caching offered by Flyweight design pattern, which makes it more efficient and useful.

Java Interview Questions from OOP and Design Patterns

74. **What is the difference between abstract class and interface in Java?**

Answer: There are various differences between abstract class and interface in Java, however, the most significant would be Java's restriction on permitting a class to extend just one class but lets it implement multiple interfaces.

An abstract class is good to define default behaviour for a family of class, but the interface is good to outline which is then used to leverage Polymorphism.

75. **Explain Liskov Substitution Principle.**

Answer: According to the Liskov Substitution Principle, Subtypes must be appropriate for super type i.e. methods or functions which use super class type must be able to work with object of subclass with no issues.

LSP is closely related to Single Responsibility Principle and Interface Segregation Principle. If a class has more functionality, then subclass might not upkeep some of the functionality and does violate LSP.

To follow the LSP SOLID Design Principle, derived class or subclass must improve functionality, but not lessen them. LSP represent 'L' on SOLID acronym.

76. **What is Law of Demeter violation and why does it matter?**

Answer: Java is centred on application programming and structuring code. If you have good knowledge of common coding best practices, patterns, and what not do, then you can write good code.

Law of Demeter suggests you 'talk to friends and not stranger', therefore used to reduce coupling between classes.

77. **What is Adapter pattern and when would you use it?**

Answer: Adapter pattern provides interface conversion. For example, if your client is using some interface but you have something else, you can write an adapter to bridge them together.

78. **What is an abstract class? How is it different from an interface, and why would you use it?**

Answer: An abstract class is a class which can have state, code, and implementation, but an interface is a contract which is totally abstract.

The abstract class and inheritance equally take precautions that most of the code is written with abstract and high-level classes, therefore it can influence Inheritance and Polymorphism.

79. **Which is better: constructor injection or setter dependency injection?**

Answer: Both have their advantages and disadvantages. Constructor injection guaranteed that class will be initialised with all its dependency. However, setter dependency injection offers flexibility to set an optional dependency.

Setter dependency injection is also more understandable if you are using XML file to define dependency. A general rule of thumb is to use constructor injection for compulsory dependency and use setter injection for non-compulsory dependency.

80. **What is the difference between Adapter and Decorator pattern?**

Answer: Though they are both similar, the difference is the intent of each pattern. The adapter pattern is used to bridge the gap in the middle of two interfaces, but Decorator pattern is used to add an extra level of indirection to support distributed, controlled or intelligent access.

81. **What is Template method pattern?**

Answer: Template pattern provides an outline of an algorithm and lets you configure or customise its steps. For example, you are able to view a sorting algorithm as a template to sort object.

It describes steps for sorting but lets you arrange how to associate them using Comparable or something comparable in another language. This pattern uses double dispatch to supplement another level of indirection.

82. **What is the difference between Inheritance and Composition?**

Answer: Both allow code reuse, however, Composition is more flexible than Inheritance because it lets you switch to a different implementation at run-time. Code written using Composition is also better and easier to test than code including inheritance hierarchies.

83. **Explain overloading and overriding in Java.**

Answer: They both let you write two methods of different functionality but with the same name, but overloading accumulates time activity while overriding is runtime activity. You can overload a method in the same class, however, you can only override a method in child classes.

It is worth noting that Inheritance is necessary for overriding.

84. **What is the difference between nester static class and top-level class?**

Answer: A public top-level class must have the same name as the name of the source file – there is no obligation for nested static class.

A nester static class is at all times inside a top-level class and you need to use the name of the top-level class to refer nested static class. For example, HashMap.Entry is a nester static class, whereby HashMap is a top-level class and Entry is a nested static class.

85. **Is it possible to write a regular expression to check if String is a number?**

Answer: A numeric String is only able to contain digits i.e. 0-9 and +/- sign. By using this information, you can write following regular expression to check if given String is number or not.

86. **What is the difference between throw and throws in Java?**

Answer: The throw is used to actually throw an instance of java.lang.throwable class, meaning you can throw both Error and Exception using throw keyword.

However, throws is used as part of method declaration and indicate which kind of exceptions are thrown by this method, so that its caller can handle them.

It is compulsory to assert any unhandled checked exception in throws clause in Java.

87. **What is the difference between Serializable and Externalizable in Java?**

Answer: Serializable interface is used to make Java classes serializable so that they can be transmitted over the network or their state can be kept on disk. However, it influences default serialization built-in JVM, which is pricey, fragile, and unsecured.

Externalizable lets you fully control the Serialization process, identify a customer binary format and enhance security measure.

88. **What is the difference between DOM and SAX parser in Java?**

Answer: DOM parser loads the whole XML into memory to create a tree-based DOM model. This helps it quickly locate nodes and make a change in the structure of XML. SAX parser is an event based parser and does not load the whole XML into memory.

For this reason, DOM is quicker than SAX but it needs more memory and is not fitting to parse large XML files.

89. **Explain 5 features introduced in JDK 1.7.**

Answer:

- try-with-resource statements free you from closing streams, and resources when you are finished with them, Java automatically closes this.
- Fork-join pool to implement something like the Map-reduce pattern in Java, which allows String variable, and literal into switch statements.
- Diamond operator for improving type inference, so there is no need to assert generic type on the right-hand side of variable declaration any longer, meaning the results are more clear and the code is more concise.
- Improved exception handling, which lets you catch multiple exceptions in the same catch block.

90. **Explain 5 features introduced in JDK 1.8.**

Answer:

- **Lambda Expression**: This allows you to pass an anonymous function as object.
- **Stream API**: Allows you to take advantage of multiple cores of modern CPU and lets you write concise code.
- **Date and Time API**: There is a solid and easy to use date and time library in JDK.
- **Extension Methods**: You can include static and default method into your interface.
- **Repeated Annotation**: This lets you apply the same annotation multiple times on a type.

91. **What is the difference between Maven and ANT in Java?**

Answer: Both are a build tool and used to create a Java application build but Maven is more advanced. It provides a standard structure for Java projects based on the 'convention over configuration' concept and routinely manages dependencies (JAR files on which your application is dependent) for Java application.

92. **What is the difference between checked and unchecked exceptions?**

Answer: Checked exception are checked at compile time. If your code throws a checked exception, it must handle it or specify it using the 'throws' keyword.

Unchecked exceptions extend the RuntimeException or Error class and do not need to be handled in the code if they are thrown or specified using 'throws'. You can always write code to specifically handle an unchecked exception.

93. **What is the difference between Object Oriented Programming and Object Based Programming?**

Answer: Object oriented programming supports all the usual OOP features such as inheritance and polymorphism. It also has no built in objects. Object based programming does not support inheritance or polymorphism and does have some built in objects.

Android

- What is the structure of an Android Application?
- What's the AndroidManifest file?
- What is an Activity in Android?
- Can you explain me how the lifecycle of an Activity works?
- What are "launch modes"? What types of launch modes are supported?
- What is a Fragment? Why are they useful?
- Are Fragments async or sync?
- Can you explain me how the lifecycle of a Fragment works?
- What's a Dialog in Android?
- What's a View in Android?
- Can you create custom views? How?
- What are ViewGroups and what's their difference from a View?
- What are the most relevant attributes from a View that you remember?
- What's an InputType?
- What is a Layout in Android?
- What different Layouts do you remember?
- What's a FrameLayout? When should you use FrameLayout instead of RelativeLayout?
- What is the resources folder and what is it used for?
- What are permissions on Android?
- What is an Intent? How many different types do you know?
- How can I persist information in an Android device?
- What is a Service in Android? How many types of Services do you know?
- What is a ContentProvider and what do you use it for?
- What is a BroadcastReceiver?
- What is ADB?

- What is DDMS and what can you do with it?
- What is an AsyncTask? Elaborate.
- What is an ANR? How can you avoid it?
- How would you implement a list in Android?
- What is the support library? Why was it introduced?
- What have you used on Android? (Activities, Fragments..)
- Why did you use Fragments?
- Why should you avoid to run non-ui code on the main thread?
- How do you run “background” code? `AsyncTask`, `Service`, `thread`? Why?
- Down sides of `AsyncTasks`?
- How did you invoke your Activity back from the `AsyncTask`?
- What kind of layouts have you developed?
- What’s different about `ConstraintLayout`?
- How did you support different types of resolutions?
- If you use include in layouts can you override properties? Can every property be overridden ? (layout_*)
- How did you communicate between fragments and activities?
- What are some problems that can arise from Fragments being async?
- Have you used storage on Android? What kind?
- What do you use `SharedPreferences` for? Why?
- What do you use database for? Why?
- Did you develop using pure SQL or an ORM/lib ?
- Do you encrypt anything you store?
- How do you test your code?
- What is Doze? What about App Standby?
- What can you use for background processing in Android?
- What is ORM? How does it work?
- What is a Loader?
- What’s `AIDL` in Android?
- What is proguard used for?

- Do you know what's obfuscation? What is it used for? What about minification?
- What is the NDK and why is it useful?
- What is a PNG9 image? What about a SVG?
- What is the StrictMode?
- What's the difference between flavors and project libraries? When would you use each?
- What is a `BuildType` in `Gradle`? What can you use it for?
- Why is it better to use `Parcelable` than `Serializable` in Android?
- What is Lint? What is it used for?
- What is a SurfaceView?
- What's the difference between `ListView` and `RecyclerView`?
- What's the `ViewHolder` pattern? Why should we use it?
- What are `DiffUtils` for (on the context of List adapters)?
- What is the XML tag `animateLayoutChanges` for?
- Is it possible to implement push notifications in Android? How?
- How can two fragments communicate?
- In what Thread does a `Service` run?
- Is a Context always from an `Activity`?
- What is a `PendingIntent`?
- What's the best way to update the screen periodically?
- Can you manually call the Garbage collector?
- What are the different types of `Broadcasts`?
- What are the difference between the different types of `Services`?
- If you have a `BroadcastReceiver` defined, the method `onReceive` will run on which thread?
- How can you communicate between an object you want and your `Service`?
- Between `binders`, `aidl`, `intents`, local and global `broadcast` what is your preference and why?

- Do you know any ORM Android libraries? What are their advantages and disadvantages?
- Have you developed widgets? Describe.
- What Android/Java libraries have you used?
- Describe the architecture of your last app.
- How do you build your apps for release? Do you have a build server?
- Have you done unit testing or automatic testing?
- How does the Android GC works? (Mark Sweep Algorithm)
- When is a object eligible for garbage collection?
- What happens if a static variable is pointing to an `Activity` `Context`? What about for the `Application` `Context`?
- What happens if you have an `Activity` that triggers a `Dialog` or a `Fragment` while stopping (after `onSaveInstance`) ?
- Could you describe what each type of context can do? For example, can I start an activity with an `Application` `Context`?
- Can you think of a limitation on Proguard? How do you overcome them?
- What are the differences between Dalvik and ART?
- Do you know what's the view tree? How can you optimize its depth?
- What is the onTrimMemory method for?
- Is it possible to run an Android app in multiple processes? How?
- Are SQL Injection attacks possible in Android? If so, how to prevent them?
- How does the OutOfMemory happens?

- What is a spannable?
- Does a fragment need a parameterless constructor? Why?
- What is renderscript? When would you use it?
- What did you did to guarantee modularity in your apps? Interfaces? Service providers? Break the project in several smaller libs? Use dependency injection? Elaborate.
- Do you know MVC, MVP…? Elaborate
- Describe the architecture of your best app. Elaborate
- Do you know Aspect-oriented-programming? What would you use it for on Android?
- What do you think is a good process to implement continuous integration/delivery?
- If we needed to build a text messaging app, please describe me the architecture you would follow
- Handlers vs Thread vs Looper
- JobSchedular Vs AlarmManager
- What is Work Manager?
- What is ThreadPoolExecutor
- Do you know about Mocks and Locks?
- Hashmap Implementation in android?
- Multitasking vs Multithreding
- How Kernel android works?
- Android Activity Memory Allocation vs Fragment Memory Allocation
- Can we handle network thread inside service?What are the pros and cons?
- Super class of Arraylist and what is the difference between ArrayList and List ?

Broadcast receivers - Interview questions:

1) what is broadcast receiver?

ans : It is a component of android, which is used to receive/ listen important system events.

eg: BATTERY LOW, POWER CONNECTED, ...

2) How do you start a receiver?

ans : By using **Intent**, and **sendBroadcast()** method

3) How do you kill a receiver?

ans : abortBroadcast() method.

4) types of receivers?

ans : static receivers, dynamic receivers, ordered receivers, sticky receivers.

6) What are the life cycle methods of broacast receiver?

ans : onreceive()

Database related interview questions:

1. What is database?

ans : It is a logical container of data/ information.

2. Which database do we use in android?

ans : SQLite - RDBMS

3. Where is database stored?

ans : Database is stored within the application, in Internal memory.

4. How to store images/ audios/ videos in a table using SQLite?

ans : by using **BLOB datatype** = binary large object .

First we have to convert images to 0101, then store in tablee using BLOB data type. Later while reading the image we will read 0101 and recreate the image by using BitMapFactory class.

5. what is primary key?

ans : unique + not null.

Primary key is used to uniquely identify each row.

6. what is foreign key?

ans : used to maintain relationship between tables. If a primary key of one tables comes to other table then it becomes foreign key.

Every RDBMS should support foreign key. But by default SQLite doesn't have support for foreign key. To enable foreign key support below command should be issued.

db.execSQL("pragma FOREIGN KEY;");

7. how to upgrade database? (imp)

ans :

a. change application version code and version name in build.gradle.

b. change database version in helper parameter.

c. go to inner helper class, go to onupgrade() method, use if-else condition and we can add/ remove/ alter tables based on the version.

8. Difference between delete and drop? (imp)

ans : delete is DML statement. drop is DDL command.

drop = delete the rows + delete the table.

delete = will delete only rows.

9. Difference between update and upgrade?

ans : update is DML statement. but upgrade is related to DDL.

update = will update the rows.

upgrade = create table/ drop table/ alter table in the future application releases.

note : upgrade terminology is used for databases.

Activity & Fragment Interview questions:

1. how do you start an activity?

ans:Using **intent** class& **startActivity()** method

2. how do you start a fragment?

ans:by using FragmentManager, fragmentTransaction & add() method.

[or]

using <Fragment ..tag> in xml.

Note : first approach is dynamic fragment, 2nd is static fragment.

3. how do you kill an activity?

ans: finish() method

4. how do you kill a fragment?

ans:

fragmenttransaction.popBackstack() method [or]

fragmenttransaction.replace() [or]

fragmenttransaction.remove()

5. how do you pass data to an activity?

ans: By using Intent & putExtra() method.

6. how do you pass data to fragment?

ans: By using Bundle, setArguments() method.

6.1: What is fragment?

def: Fragment is a "re-usable UI component". Fragments are introduced in android 3.0. Fragments are used to design UI for mobiles & tablets.

7. User clicks back button, which life cycle methods will be called for activity?

ans : onpause, onstop, ondestroy.

8. User clicks home button, which life cycle methods will be called for activity?

ans : onpause, onstop

9. User is looking at screen & suddenly call comes which life cycle methods will be called for an activity?

ans : onpause, onstop

10. User rotates the phone, life cycle methods of an activity? [IMP]

ans : onpause, onsaveinstancestate, onstop, ondestroy,

oncreate, onstart, onrestoreinstancestate, onresume.

same for virtual kyeboard up/down & language changes also. same for low memory also.

11. write one line about each life cycle method.

oncreate()

a. first life cycle method

b. we will load screen design here

c. we will initialize all views here

onapuse()

a. when user is moving away from the screen this method will be called.

b. eg: popup displayed, new screen comes, etc..

c. important data has to be saved here. eg: saving database, saving files etc..

onsaveinstancestate()

a. this is called in configuration changes

b. eg: phone rotation, language changes, etc.

c. programmer has to save activity states here and restore in onRestoreinstancestate() method.

12) What is the use of frgment?

ans: For designing applications for mobile phones & tablets, we use fragments.

Material Design Api classes & definitions:

CoOrdinatorLayout :

a. It is inherited from FrameLayout

b. It is used to place elements on one-on-top of other, with shadow/3d effect.

c. If coordinator layout is removed, then 3d layering will not happen properly.

AppBarLayout:

a. It is used to club ActionBar+Tabs.

b. It is used to have parallox effect on actionbar. Parallox effect means - when user scrolls up / down, then actionbar will be moved out of screen.

ToolBar :

a. it is introduced in android 5.0

b. it is actionbar with shadow effect.

c. toolbar can be placed anywhere on the screen.in olden days, action bar used to stick at top.

TabLayout :

a. it is used mostly with viewpager.

b. it inherits from horizontal scrollview.

c. mostly it is attached to actionbar/toolbar.

FloatActionButton:

a. it appears at right corner of the screen.

b. Most important item should be placed as FAB.

eg: gmail app places "compose mail" as FAB.

c. this is highest priority icon.

NagivationView :

a. It is used to attach sliding menu with actionbar.

b. it will have header and menu body portions.

SnackBar :

a. it is advanced version of Toast.

b. it comes with colors & buttons.

c. it can be swipe deleted. [not possible with toast

RecyclerView :

a. Advanced version of listview.

b. it is faster & memory efficient compared to list view.

c. It comes with viewholder design pattern.

d. can be used as listview/ gridview/ staggered grid

CardView :

a. it inherits from framelayout.

b. it is is used to design round cornered - cards, with 3d shadow effect.

c. mostly used along with recyclerview - to show good looking rows.

Network related questions:

1. what is URL api?

a) it is used to represent website url.

2. what is HttpUrlConnection api?

a) it is used to establish connection between android application & server.

3. what is InputStream api?

a) this API is used for GET request. if we want to read some data from server, we will use this API. this is where the actual data will come from server.

Note : this may take some time, depending on internet connection.

4. what is InputStreamReader API?

a) it is used to convert/read the raw data [bits] coming in the input stream.

note : this api converts bits to ascii chars.

note : it is not efficient as it reads each character one by one.

5. what is BufferedReader API?

a) this api provides an efficient way to read data coming from server in big chunks, in line by line fashion.

Note : InputStreamReader vs BufferedReader is interview question.

Terminology used in websites:

Server/Webserver : it is combination of hardware & software where websites [website code] will run.

eg : apache webserver, tomcat webserver - for J2EE websites.

IIS webserver - for .NET websites.

webconfig : it is a small xml file in the server, which is similar to our android manifest xml file. this xml file contains, which URL is mapped to which Servlet this xml file also contains, which URL is mapped to which webservice.

Servlet : full form is serving the request.

Servlet is a small java file, which sits in server.

Servlet class will generally extend a predefined class HTTPServlet.

Servlet will handle all incoming requests from BROWSER.

Servlet is the first entry point in the server, which handles incoming request.

JSP : Java server pages. It is a small java file which sits in server. Main responsibility of JSP class is to prepare HTML content.

MVC : Model View Controller, is a design pattern used while designing web sites. Controller -is- servlet. View -is- JSP/ JSF. Model -is- Database connection logic.

WebServices: It is a small java file which sits in server.

Webservices will handle all incoming requests from non-browser applications.

Eg: if an android application sends a request to server, it most likely hits a web service.

Note : A website can have n-num of servlets/ webservices.

Note : Each servlet/ webservice will have one URL.

Types of webservices: RESTful & SOAP

RESTful Webservices: Representational State Transfer web services. Faster than SOAP services. Most of the RESTful services uses JSON format to transfer the data.

SOAP: Simple object access protocol. Slower compared to REST. Mostly SOAP services uses XML format to transfer the data.

HTTP : is a common language/ protocol used over internet for communicating between client & server systems.

HTTPRequest : it is a class representing request going from client to server.

HTTPResponse : it is a class representing response coming from server to client.

interview questions on services

1. what is service?

ans : component android used to do background task.

eg: connecting to internet.

2. types of services?

ans : Service & IntentService

3. service life cycle methods?

ans : oncreate, onstartcommand, ondestroy, onbind.

4. what is intent service?

ans : Service with 1 background thread.

lifecycle methods - constructor, onhandleintent

5. how to kill a service?

ans : stopservice(intent);

6. what is a thread?

ans : thred is a light weight process [or]

thread is an independent path of execution.

7. thread vs service?

ans : thread is os component, service is android component

interview questions on services - part 2

1. what is ANR?

ans : Application Not Responding is a famous error

in android.

2. why ANR error will occur?

ans : if key events are not delivered in 5 seconds time limit, then O.S will show this error popup to user.

3. What is the root cause for ANR errors?

ans : if service is not having its own thread, then service will use main thread for doing background task. when main thread is busy in the service, it can't handle key events. That leads to ANR.

4. what precautions to take to avoid ANR errors?

ans : when we create a service, create a thread also in the service.

Note : 1 thread handling too many taks is not good design.

Activities - will always use - MainThread.

Services - should always use - separate thread.

5. Some other scenarios where ANR may occur?

ans : opening, inserting, reading database in activity is danger. It might lead to ANR.

Activities - will use - main thread.

database - also using - main thread.

6. what is the time limit of ANR?

ans : 5 seconds. [set by google]

but phone manifacturer can reduce it for more efficiency. but should not increase it.

Shared preferences

1) What is preferences/ sharedPreferences?

ans : It is a small xml file which is used to store small amount of data permenantly.

eg: Storing user registration details/ credentials.

Note : we can only store primitive data types.

2) Which API we have to use to create a preference file?

ans : getSharedPreferences()

3) What is the 2nd parameter while creating

new preference file?

ans : 1st - file name, 2nd - PRIVATE MODE [0]

4) Which API we have to use to open preference file?

ans : getSharedPreferences()

5) Can I store array of strings in a preference file? how?

ans : putStringSet() - convert array to set and store.

6) How do you save a preference file?

ans : apply() [or] commit()

7) What is the difference between commit() & apply().

ans : commit() is slow, because it runs on main thread. apply() is fast, as it runs on different thread.

8) Is preference files secured?

ans : yes, but we have to use 2nd parameter 0 - PRIVATE MODE

9) How do you store multiple user credentials using preferences?

ans : op1 : for each user create a seperate pref file.

op2 : use 1 pref file but use different keys/tags for each user -

et.putString("user1",...);

et.putString("user2",...);

10)List all the methods of SharedPreferences class?

ans : getInt, getstring, getboolean, getfloat, getlong, getStringset

11)List all the methods of Editor class?

ans : putint, putstring, putboolean, putfloat, putlong, putstringset, apply, commit

12)How to create a preference file without name?

ans : SharedPreferences sp = getSharedPreferences("abc",0);

SharedPreferences sp = getPreferences(0);

13)what is the extension of preference file?

ans : xml

14)Can other applications access our preference file?

ans : generally no.

a. if other app has root permission

b. if second parameter is 1/2 [world readable/writab]

15)Can other activity of same application access pref file?

ans : yes

Interview questions on starting screens:

1. How do you start other activity/ screen?

ans : Intent class, startActivity() method.

2. How do you pass data from one screen to other screen?

ans : putExtra() method.

3. What is Bundle?

ans : Bundle is a container which carries data.

4. How do you kill a screen?

ans : finish() method.

Interview questions on passing data:

1. How do we pass data from one screen to other screen?

a) using Intent class, & putExtra() method.

2. Is there any other way to pass data to other screen?

a) yes, Serialization. [For passing objects]

note : Serialization also uses intent & putextra.

3. What is Serialization?

a) The process of converting "java object -to- bits & bytes".

4. what is De-Serialization?

a) The process of converting "bits & bytes -to- java object".

5. what is Serializable?

a) It is a marker interface [empty interface] in JDK s/w.

6. What is difference between Serializable vs Serialization?

a) Serializable is predefined interface [empty] of java. Serialization is a process of converting "java obj to bits". Serialization is done by JVM.

7) What is Parcellable?

a) It is predefined interface of Android.

8) What is parcelling? [or] what is the use of Parcellable?

a) Serialization is a java technique, which takes more memory and it is slower. Parcellable is an android technique, to pass objects faster and with less memory wastage.

Company interview questions - part 1

1. DIF between ART & DVM

ans : ART is replacement for DVM, in android 5.0

ART is 4times faster than DVM.

ART uses Ahead of Time compiler, DVM uses JIT compiler

2. what is dx file, how it is generated?

ans : dx = dalvik executeable file.

generated by build.gradle.

apk = android application package file.

generated by apkgen tool.

3. what is intent-filter?

4. what is broadcast receiver?

ans : it is a component of android which is used to listen or receive important system events.

ways : static - through xml

dynamic - through java code

5. where will you catch announcements / system events.

ans : using broadcast receiver.

6. what is content provider?

ans : it is used to share data from one application to other application.

eg : contact application is sharing numbers to whatsp

7. is there any other way to share data between 2 apps.

ans : using server. [or] cloud.

8. service life cycle methods?

ans : oncreate/ onstartcommand/ ondestroy.

9. types of services?

ans : service & IntentService.

10. when to service & when to use intentservice?

ans : Service does not come with default thread. Service depends on Main thread. Service has 4 life cycle methods. Using only service will lead to ANR error. Service need to be stopped explicitly either by stopserver [or] stopselfresult() methods.

IntentService class inherits from Service class. IntentService class comes with one thread. Using IntentService solves ANR problem. IntentService will kill itself when there are no more requests to start the service.

Some java related questions asked for android engineers:

1. dif between abs class & interface.

ans : abstract class can have abstract methods & concrete methods. abstract class is partial abstraction. Abstract class should be extended. interface can have only method declarations. interface is used to achieve pure abstraction. interface should be implemented.

2. dif bw hashmap & hashtable?

ans : HashMap part of collection framework library.jdk1.2

HashTable is legacy class. JDK 1.0

HashMap allows 1 null key, but HashTable will not.

HashMap is not synchronized, HashTable is synchronizd

Some more android questions

1. what is intent? types of intent?

ans : Intent is predefined class which is used to start other activities.

types : implicit, explicit, pending, sticky.

2. how will you start activity if you are expecting data back?

ans : startactivityforresult(in, requestcode).

3. what is child activity?

ans : Each screen in android is called as an activity.

4. can we use intent to start content provider?

ans : using intent you can start activity/service/receiver. we can't start content providers using intent. We use ContentResolver to start content provider.

5. does java support multiple inheritance?

ans : no for class, yes for interfaces. because it leads to function ambiguity.

6. how many ways we can store data in android. What all the different data storage options available in android?

ans : preferences[offline], sqlite db [offline], server[online], cloud[online], sd card[offline], internal memory[offline].

7. what manifest file contains?

ans : <manifest>

<uses-permission>

<application>

<activity>

<intent-filter>

8. types of access modfiers?

ans : private, default, protected, public.

9. can we access protected variables in child class?

ans : yes.

10. string vs stringbuilder vs stringbuffer

ans : Strings are immutable, stored in s.c.pool

stringbuilder is mutable, stored in heap, not sync

stringbuffer is mutable, stoerd in heap, synchronized

Interview questions part 2:

1. What is activity & write activity life cycle methods?

ans : Each screen in android is an Activity.

2. what is fragment & write fragment life cycle methods?

ans : fragment is a modular & reusable UI component.

[or]

fragment is used to design screens/applications for

mobles & tablets.

3. write string polyndrome logic?

ans :

4. User is going back from second activity to main activity, draw life cycle methods?

ans : onrestart, onstart, onresume

5. what is recyclerview?

ans : RecyclerView is introduced in MaterialDesign [5.0]

RecyclerView is faster than ListView, because of

ViewHolder design pattern.

6. how to pass data from one activity to other?

ans : intent, putextra.

7. what is intent filter?

ans : <intent-filter> tag will be in manifest file.

use case not yet covered. will be covered later.

8. what is broadcast receiver? can we create custom

broadcast receiver?

ans : it is component of android. it is used to receive important system events.

eg : I want to listen charger plugged.

eg : I want to listne when battery low.

9. what is asynctask? and draw life cycle methods of async task?

ans : It is used to create threads in android application.

onpreexecute, doinbackground, onprogressupdate, onpostexecute.

10.What are the components of android?

ans : activity, service, Broadcast Receiver, ContentProvider.

11.what is database.

ans : Database is a container of data in table format.

12.what is service? how many types of services are there?

ans : Service is a component of android, Which is used to do background task.

types : Service and IntentService.

Interview questions part-3:

1. What is activity & Intent?

ans : Each screen in android is an Activity.

Intent is a predefined class of android, which is used to start other activities & pass data.

2. Difference between activity & Intent?

ans : Activity is a component of android, it has life cycle methods. Intent is a predefined class to start activities.

3. Explain different types of intents?

ans : 4.

Implicit Intent, Explicit Intent, Pending Intent,

Sticky Intent.

4. What is the use of ViewGroup?

ans : ViewGroup is a predefined class extending from View class

It is used to design screens.

Eg of viewgroups - LinearLayout, FrameLayout, RelativLayout, AbsoluteLayout is deprecated.

5. What are different types of notifications?

ans : will be covered later

6. What is GCM notification & what is the Use?

ans : will be covered later

7. What is the us of 9-Patch image?

ans : 9-patch images are scalable, without loosing resolution.

nine patch image extension is .9.png

draw9patch.exe file is used to generate 9 patch image.

8. What is launcher activity?

ans : It is the starting screen that gets started, when user opens the applicaiton.

10. How to set first/ default screen?

ans : by going to manifest file, use <intent-filter> with

<category .. LAUNCHER> & <action...MAIN>

9. How to change application language based on phone location?

ans : will be covered later.

11. what is the use Manifest file?

ans : Without manifest, our application & activities will

not start.

12. Tell me about Build.gradle?

ans : already covered.

Interview questions part-4:

1. what is android?

ans : it is an o.s for mobiles and tablets [or]

it is a software which contains 4 layers.

os/ libraries/ framework/ applications.

2. what is a fragment?

ans : fragment is a modular & reusable UI component.

[or]

fragment is used to design screens/applications for

mobles & tablets. i.e for portatit mode & landscape

mode.

3. what is a recycler view?

ans : RecyclerView is introduced in MaterialDesign [5.0]

RecyclerView is faster than ListView, because of

ViewHolder design pattern.

4. how will you pass the data?

ans : Using intent & putextra() - between activities.

[or] bundle & setarguments() - activity to fragment.

5. what is bundle?

ans : bundle is a container which stores data in key,val pairs

we will use Bundle in onsaveinstancestat &

onrestoreinstancestate, when user rotates the phone.

6. what is manifest file?

ans : this file contains <manifest> tag, <application> tag

<activity..> tags.

this file also has <intent filter>

this file also has <uses-permission > tag.

it talks about application components.

7. what is build gradle?

ans : It is a tool/program, which comes along with android

studio, which generates apk file.

In build.gradle, in dependencies section we will

keep libraries used in the application.

eg : YouTubeAndroidPlayerApi.jar

compile 'com.support.recyclerview:26+'

compile 'com.support.cardview:26+'

...material design...

...google maps...

...appcompat v7...

8. what is zomato app?

ans :

9. how do you design 20% overlapping 2 images?

ans : framelayout

10. what are the methods of asynctask? [100%]

ans : onpreexecute, doinbackground, onprogressupdate,

onpostexecute.

11. what is joins?

ans : outer join, inner join, self join.

12. what do you do onpostexecute?

ans : .....

13. how do you use http request/ http response?

ans : .....

14. when I open application, how many life cycle

methods will be called?

ans : oncreate, onstart, onresume.

15. when we press home button what happens?

ans : onpause, onstop will be called.

note : activity is not destroyed, it is still in memry.

Reactive/RxJava (2) Experience

- What's the difference between a `Observable` and a `Single`?
- What about a `Flowable`?
- What is `subscribeOn` used for?
- What about `observeOn` ?
- What are the different `subscribeOn` and `ObserveOn` pools?
- What operations do you use the most?
- What is the difference between `Observable.merge` and `Observable.combineLatest` ?
- What's the difference between a `.map` and a `.flatMap`?
- What would you use a `.filter` for?
- What's a `BehaviorSubject`? What about a `PublishSubject`?
- What happens if you have multiple `subscribeOn`? And `ObserveOn`?
- Can you have multiple subscribers for the same stream?

- What is `.share()` for?
- Do you need to dispose of streams? When?
- How can you create your own `Observable`?
- What the difference between: `doOnSubscribe`, `doAfterTerminate` and `doFinally`?
- What happens if you add a `Disposable` to a `CompositeDisposable` that has already been disposed?
- In what cases would you need to use a `Flowable`?
- What is `flattenAsObservable` for?
- How can you define your own thread pool for each `Scheduler`?
- Why would you want to define your own thread pool for a `Scheduler`?
- How would you implement a flow where when a `.flatMapIterable` item fails, the flow continues with the next without failing them all?

Exercises

- Write a method that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”
- Write a method that receives two strings, needle and haystack and without using .contains or .indexOf searches for the string needle on the string haystack and returns its index if it exists.
- Write a Java method to print Fibonacci series up to 100.
- Write Java method to reverse String in Java without using indexOf or contains.

- Write a Java method to find if a number is prime number or not
- How to find if a linked list contains cycle or not in Java
- Write a Java method to calculate Factorial of a number in Java
- Write a method that receives a number and prints the sum of all odd numbers until that one.

Referenfce:

#Medium,Techgig,Github,HackerRank,Geeks for Geeks,Google etc.

Other Android Resources.