

[Data Science](#) [Data Science Projects](#) [Data Analysis](#) [Data Visualization](#) [Machine Learning](#) [ML Projects](#) [Deep Learning](#)

Top Generative AI Interview Question with Answer

Last Updated : 23 Jan, 2025

Welcome to the Generative AI Specialist interview. In this role, you'll lead innovation in AI by developing and optimising models to generate data, text, images, and other content, leveraging cutting-edge technologies to solve complex problems and advance our AI capabilities.

In this interview, we will assess your expertise in generative models, including GANs, VAEs, attention mechanisms, and diffusion models. We seek insights into your hands-on experience, problem-solving approach, and ability to innovate and adapt in a dynamic environment.

Generative AI Interview Question with Answer

We will discuss your technical knowledge and skills, such as your familiarity with model architectures, training techniques, and evaluation metrics.

Additionally, we will delve into your practical experience with deploying and integrating generative models in real-world applications, and your strategies for addressing performance and quality issues.

Q1: What is the Architecture of Generative AI ?

Answer: Generative AI works through the use of neural networks, specifically Recurrent Neural Networks (RNNs) and more recently, Transformers. Here's a simplified breakdown of how it functions:

- **Data Collection:** To begin, a substantial amount of data related to the specific task is gathered. For instance, if you want to generate text, the model needs a massive text corpus to learn from.
- **Training:** The neural network is then trained on this data. During training, the model learns the underlying patterns, structures, and relationships within the

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

• **Generation:** Once trained, the model can generate content by taking a seed input and predicting the subsequent elements. For instance, if you give it the start of a sentence, it can complete the sentence in a coherent and contextually relevant manner.

• **Fine-Tuning:** Generative AI models can be further fine-tuned for specific tasks or domains to improve the quality of generated content.

Q2: What are the top applications of Generative AI?

Answer: Generative AI has a wide range of applications across different industries:

- **Natural Language Processing (NLP):** It's used for text generation, language translation, and chatbots that can engage in human-like conversations.
- **Content Generation:** Generative AI can create articles, stories, and even poetry. It's used by content creators to assist in writing.
- **Image and Video Generation:** It can generate realistic images and videos, which are valuable in fields like entertainment and design.

Q3: . How is Generative Adversarial Networks (GANs) used in AI?

Answer: Generative Adversarial Networks, or GANs, are a type of machine learning model made up of two key parts: **a generator and a discriminator**. These two components collaborate like — the generator creates new content, while the discriminator evaluates it, pushing both to improve through their interaction.

Q4: How Generator and Discriminator works together ?

The generator tries to fool the discriminator, while the discriminator aims to correctly classify the data. They improve through this competition until the generator produces highly realistic outputs.

Analogy: Counterfeitors and Currency Experts

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

bills that look and feel exactly like real currency.

-
- **Discriminator:** Think of a team of currency experts at a bank. Their job is to examine money and determine if it's authentic or counterfeit.

Q5: What are some common techniques to improve the stability and performance of GANs?

Answer: To enhance the stability and performance of GANs, various strategies can be employed. Using architectures like **Deep Convolutional GANs (DCGANs)** or **Wasserstein GANs (WGANs)** is a popular approach. Techniques such as **batch normalization, feature matching, and gradient penalty** are effective in stabilizing training. Additionally, leveraging advanced optimizers and custom loss functions tailored for GANs can further boost their performance.

Q6. What is Deep Convolutional Generative Adversarial Networks (DCGANs) ?

Answer: Deep Convolutional Generative Adversarial Networks (DCGANs) are a type of generative adversarial network (GAN) architecture that leverages convolutional neural networks (CNNs) for both the generator and discriminator components.

It includes replacing pooling layers with strided convolutions in the discriminator and fractional-strided convolutions in the generator.

Suppose you train a DCGAN on a dataset of handwritten digits (like MNIST). After training, the generator can produce entirely new, realistic-looking handwritten digits that resemble the style of the dataset but are unique and not present in the original data. For instance, the generator might create a "handwritten 7" that looks like it was written by a human but was entirely generated by the network.

Q7: What are the key differences between conditional GANs (cGANs)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Answer: Conditional GANs extend traditional GANs by conditioning the generation process on additional information, such as class labels to generators. In contrast, traditional GANs generate data without any conditioning.

Conditional GANs (cGANs) are like ordering a custom cake. Instead of just asking for a cake (like in regular GANs), you give the baker specific instructions: "make me a chocolate cake with red velvet frosting and sprinkles." These instructions (called conditions) tell the GAN to generate data that meets those specific requirements. This extra information gives you more control over the generated output, allowing you to create images with specific objects and attributes, music in a particular style, or text with a desired theme.

Q8: Can you explain the purpose and implementation of regularization techniques in generative models?

Answer: Regularization techniques are essential for preventing overfitting and improving a model's ability to generalise to new data. In generative models, common methods include dropout, weight decay, and spectral normalization. These methods ensure that the model learns robust features and avoids memorizing the training data.

- **Dropout** randomly disables some neurons during training, forcing the model to rely on multiple features.
- **Weight Decay** penalises large weights, ensuring simpler and more general features are learned.
- **Spectral Normalization** stabilises training by limiting the largest singular value of weight matrices.

Q9: Explain the concept of a Variational Autoencoder (VAE).

Answer: A Variational Autoencoder (VAE) is a machine learning model that takes data (like an image), compresses it into a small set of numbers, and then recreates the original data from those numbers.

allows it to generate new data by sampling these numbers, creating outputs similar to the original examples.

Essentially, a VAE learns patterns in the data and uses them to create unique but similar outputs.

Q10: What distinguishes VAEs from ordinary Autoencoders?

Answer: Autoencoders compress data into a fixed set of values and reconstruct it exactly, using a single point in the latent space. In contrast, Variational Autoencoders (VAEs) treat the compressed data as a range of possible values, represented by a mean and variance, and sample from this range. This probabilistic approach allows VAEs to generate new data by sampling from these ranges, making them more suitable for creating new examples than standard autoencoders.

Imagine you're describing a friend's personality:

- **Autoencoder:** You give a single, precise description like "friendly and outgoing." It's a deterministic summary, capturing the essence but lacking nuance.
- **VAE:** You describe your friend as "generally friendly" (mean) but sometimes "a bit shy" (variance). This captures the variability and range of their personality, allowing for a more nuanced and realistic understanding.

Q11: How do GANs differ from VAEs?

Answer: VAEs: Learn a latent space distribution, optimizing a reconstruction + objective. They produce smooth latent spaces and probabilistic generation, but may produce blurrier samples.

- **GANs:** Use an adversarial loss, often producing sharper, more realistic outputs. However, they may lack explicit latent variable inference and stability of training can be more challenging.

Q12. How do Diffusion Models generate images?

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Diffusion models gradually **add noise to data and learn to reverse this process, denoising step-by-step to generate samples.**

Diffusion models, like DDPM and Stable Diffusion, create high-quality and diverse images by learning to gradually remove noise from data. This step-by-step denoising process helps them generate realistic images, making them popular for tasks like image creation.

Q13. What is Stable Diffusion and why is it significant?

Stable Diffusion is a popular text-to-image diffusion model that generates high-quality, diverse images from text prompts. It can run efficiently on consumer GPUs due to latent diffusion techniques.

Being open-source made advanced image generation widely accessible, encouraging innovation and contributions from the community

Q14. Explain Latent Diffusion Models.

Latent Diffusion Models apply the diffusion process in a latent space (from a pretrained autoencoder) rather than on raw pixels. This reduces computational complexity and speeds up sampling. By working in a lower-dimensional latent space, they preserve fidelity and detail while enabling powerful capabilities like text-conditioned image generation.

Q15: How do you evaluate the performance of a generative model, and what metrics are commonly used?

Answer: The performance of generative models is evaluated using a combination of quantitative metrics and qualitative assessments.

Common metrics include **Fréchet Inception Distance (FID)**, **Inception Score (IS)**, and **BLEU score**. Qualitative evaluation involves human judgment to assess the relevance of generated samples. Combining these methods provides a comprehensive view of model performance.

Q16: Explain the basic difference b/w Fréchet Inception Distance (FID), Inception Score (IS), and BLEU score for text generation.

Answer: FID, IS, and BLEU are metrics used to evaluate the quality of generated data. FID measures the similarity between real and generated images by comparing their feature distributions. IS evaluates the quality and diversity of generated images. BLEU measures the quality of machine-translated text by comparing it to human-translated references.

- **FID:** Compares generated images to real images.
- **IS:** Evaluates the quality and diversity of generated images.
- **BLEU:** Evaluates the quality of machine-translated text.

Q17: How do Large Language Models (LLMs) relate to Generative AI in text?

Answer: LLMs (e.g., GPT-3, GPT-4) are large transformers trained on massive text corpus. They learn complex language patterns and can generate contextually rich text. As generative AI models for language, they produce essays, code snippets, answers, and creative writing, enabling chatbots, assistants, and content generation at scale.

Q18: What is Prompt Engineering in Large Language Models?

Answer: Prompt engineering involves crafting the input text (prompts) to guide LLMs toward desired outputs. By carefully choosing words, instructions, or examples, users can influence the model's style, format, and correctness. Effective prompts can drastically improve results without changing the model's parameters.

Q19: Explain Few-Shot and Zero-Shot learning in LLMs.

Answer:

Zero-Shot: The model performs a new task without explicit training examples

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

- **Few-Shot:** Minimal examples (e.g., a few input-output pairs) are provided in the prompt.

Imagine you're learning a new board game:

- **Zero-Shot:** You've never seen this game before. You read the rules and try to play based on your general knowledge of games (like how to take turns, the concept of winning).
- **Few-Shot:** You've never seen this game before, but someone shows you a few rounds of play. You use these few examples, combined with your general game-playing knowledge, to figure out how to play.

Q20: How do Retrieval-Augmented Generation (RAG) models improve factual correctness?

Answer: RAG models enhance the performance of traditional generative models by integrating retrieval techniques with generative capabilities. This combination allows RAG models to access external knowledge sources, significantly improving the factual accuracy of their outputs.

By grounding answers in real data, RAG ensures better factual accuracy and up-to-date information.

Practical Experience and Application

Q21: Describe a project where you applied generative models to solve a real-world problem. What was the outcome?

Answer: In a project aimed at enhancing customer support, I applied a generative model to create a chatbot that could generate contextually relevant responses. By training the model on historical interaction data, we improved response accuracy and user engagement. The chatbot significantly reduced the need for human intervention and increased customer satisfaction through more natural and personalized interactions.

Answer: Handling data privacy involves several practices, including data anonymization, encryption, and ensuring compliance with regulations such as GDPR or HIPAA. I implement techniques to de-identify personal information and use secure data storage and access controls. Additionally, I adopt privacy-preserving techniques like differential privacy when training models on sensitive data.

Q23: What are some challenges you have faced when deploying generative models in production environments, and how did you address them?

Answer: Challenges in deploying generative models include managing computational resources, ensuring real-time performance, and integrating with existing systems. To address these, I optimized model inference through techniques like model quantization and pruning, used scalable cloud infrastructure, and developed efficient APIs for integration. Regular monitoring and performance tuning were also essential for maintaining deployment efficiency.

Q24: How have you used generative models to create synthetic data, and what advantages does this approach offer?

Answer: Generative models can be used to create synthetic data for various purposes, such as augmenting training datasets or generating samples for testing. For example, I used GANs to generate synthetic medical images for training diagnostic algorithms. The advantages include increased data diversity, enhanced model training, and the ability to generate data where real samples are scarce or difficult to obtain.

Q25: Describe your experience with integrating generative models into user-facing applications. What considerations are important?

Answer: Integrating generative models into user-facing applications involves

maintaining output quality, and providing an intuitive user interface. I also ensure that the integration supports scalability and handles varying user interactions effectively.

Q26: Can you explain how you have used generative models for content creation, such as generating text, images, or music?

Answer: I have used generative models for content creation in various ways, such as generating personalized marketing content using language models, creating artwork with GANs, and composing music using models trained on MIDI data. These applications involved fine-tuning models on specific datasets to generate high-quality and contextually appropriate content that met the needs of the target audience.

Q27: What strategies do you use to ensure the quality and diversity of data generated by your models?

Answer: To ensure quality and diversity, I use techniques such as incorporating diverse training data, employing regularization methods, and using evaluation metrics that measure both fidelity and variety. For instance, I might use [data augmentation](#) techniques and introduce controlled noise to encourage the model to explore different regions of the latent space. Regular evaluation and feedback loops help refine the generated data and maintain high standards.

Q28: How do you approach the challenge of evaluating generated content from a user experience perspective?

Answer: Evaluating generated content from a user experience perspective involves gathering user feedback through surveys, usability tests, and direct interactions. I assess factors such as content relevance, coherence, and user satisfaction. Additionally, conducting [A/B testing](#) with different versions of generated content helps determine which outputs provide the best user experience and meet specific user needs.

Q29: Can you discuss your experience with transfer learning in the context of generative models?

Answer: [Transfer learning](#) involves using a pre-trained model as a starting point for training on a new task. In the context of generative models, I have used transfer learning to use existing models trained on large datasets and fine-tune them for specific applications. This approach can significantly reduce training time and improve performance, especially when working with limited data.

Q30: What considerations are involved in selecting the right architecture for a specific generative modeling task?

Answer: Selecting the right architecture involves considering factors such as the type of data like images and text, the desired output quality and the computational resources available.

For example, convolutional architectures may be preferred for image generation, while transformers might be suitable for text. Additionally, the complexity of the task, the need for conditional generation, and the specific requirements of the application all play a role in architecture selection.

Q31: Can you describe a scenario where you've used LangChain for chaining LLM tasks?

Answer: Imagine you want to build a chatbot that answers questions about a company's products.

- **Gather documents:** Collect product descriptions, manuals, etc.
- **Process documents:** Split them into smaller chunks and create a vector store to quickly find relevant information.
- **Chain LLMs:** Use [LangChain](#) to train.
- **Find relevant information:** Retrieve the most relevant document chunks for a given user question.
- **Summarize:** Have one LLM summarize the retrieved information

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Q32: What are the key differences between LangChain, Llamaindex, and Chainlit and when would you use each?

Answer:

- **LangChain:** A powerful toolkit for building diverse applications using [Large Language Models \(LLMs\)](#). It allows you to connect LLMs with other tools and data sources to create complex and interactive systems.
- **Llamaindex:** Specifically designed for applications that need to retrieve and process information from large datasets before feeding it to an LLM. This is crucial for building accurate and informative applications
- **Chainlit:** Focuses on making it easier to build and deploy user-friendly LLM applications. It helps you create intuitive interfaces and simplifies the process of getting your applications up and running.

Q33: What role do vector databases like Pinecone or FAISS play in generative AI applications?

Answer: [Vector databases](#) like Pinecone and FAISS are crucial for generative AI applications, especially those involving large datasets and complex relationships. They efficiently store and retrieve vector representations of data (like text, images, or audio), enabling tasks like RAG (Retrieval Augmented Generation), Personalised Recommendations etc.

Problem-Solving and Innovation

Q34: Describe a time when you had to innovate to solve a unique problem in generative modeling. What was your approach?

Answer: In a project where traditional generative models struggled to produce realistic 3D objects, I innovated by combining GANs with geometric constraints to guide the generation process. I integrated a 3D-aware discriminator that evaluated the geometric properties of generated objects, leading to more realistic and usable results. This approach involved designing new loss

Q35: How do you approach debugging and optimizing the performance of generative models that are underperforming?

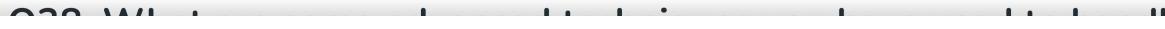
Answer: Debugging and optimizing underperforming generative models involve a systematic approach. I start by analyzing the model's outputs and training logs to identify issues. I review the data quality, training procedures and model architecture. Techniques such as hyperparameter tuning, model simplification, and applying advanced regularization methods can help address performance issues. Additionally, conducting ablation studies to test the impact of different components can provide insights into potential improvements.

Q36: Can you provide an example of how you've leveraged ensemble methods in generative modeling?

Answer: I have used ensemble methods in generative modeling by combining multiple models to improve overall performance. For instance, in image generation tasks, I combined outputs from several GANs trained with different architectures and hyperparameters. This ensemble approach helped mitigate individual model weaknesses and produced higher-quality and more diverse generated images. Combining predictions from multiple models can enhance robustness and accuracy.

Q37: How do you balance creativity and coherence in the outputs of generative models?

Answer: Balancing creativity and coherence involves setting appropriate model parameters and loss functions that encourage both novelty and relevance. Techniques such as incorporating diversity-promoting objectives, adjusting the temperature in sampling strategies, and using evaluation metrics that assess both creativity and coherence help achieve this balance. Regular feedback and iterative refinement based on user or expert evaluation also play a crucial role in maintaining this balance.



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Answer: To handle mode collapse in GANs, I have employed techniques such as using a variety of discriminator architectures, incorporating feature matching loss, and applying historical averaging. Techniques like mini-batch discrimination and unrolled GANs can also help address mode collapse by encouraging the generator to produce diverse outputs and avoid overfitting to a limited set of modes.

Q39: How do you approach the problem of generating high-quality samples in the presence of noisy or incomplete training data?

Answer: Generating high-quality samples with noisy or incomplete data involves employing robust training techniques and data preprocessing. I use data augmentation and denoising methods to improve data quality and robustness. Additionally, I may employ regularization techniques and advanced architectures that are less sensitive to noise. Techniques like adversarial training can also help improve the model's ability to generate high-quality samples despite data imperfections.

Q40: Can you discuss how you have used reinforcement learning in conjunction with generative models to enhance performance?

Answer: I have used reinforcement learning to fine-tune generative models by defining specific reward functions that guide the generation process. For example, in text generation tasks, reinforcement learning was employed to optimize the model based on user engagement metrics or content relevance. This approach involves training the model with a reinforcement learning algorithm to improve performance based on feedback from the environment or user interactions.

Q41: What strategies do you use to ensure the generated outputs are aligned with ethical guidelines and avoid unintended consequences?

Answer: Ensuring alignment with ethical guidelines involves implementing

bias, and establish clear guidelines for acceptable use. Additionally, involving diverse stakeholders in the development process and implementing robust monitoring and feedback mechanisms help address ethical concerns and avoid unintended consequences.

Q42: How do you approach the challenge of integrating generative models with existing machine learning pipelines and systems?

Answer: Integrating generative models with existing machine learning pipelines involves designing compatible interfaces and ensuring seamless data flow between systems. I focus on creating APIs and data pipelines that facilitate smooth interaction between models and other components. Additionally, I consider the impact on system performance and scalability, ensuring that the integration does not compromise the efficiency or reliability of the overall pipeline.

Q43: Can you describe an instance where you had to adapt a generative model to work within constraints such as limited computational resources or real-time requirements?

Answer: In a project with limited computational resources, I adapted the generative model by simplifying its architecture and applying model compression techniques such as pruning and quantization. To meet real-time requirements, I optimized the inference process using efficient algorithms and hardware acceleration. These adaptations allowed the model to operate within the resource constraints while maintaining acceptable performance levels.

Q44: What are some common challenges when deploying LLMs in production, and how do you address them?

Answer: LLMs face challenges like latency, memory usage, and scalability. To address these, techniques like:

- **Quantization:** Reducing model size for easier deployment.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

- **Cloud Solutions:** Leveraging scalable cloud infrastructure. These approaches improve efficiency and enable wider LLM adoption.

Q45: How do you ensure the ethical and secure use of LLMs in applications?

Answer: Ethical considerations include generating biased or harmful content. Addressing these involves thorough testing, filtering generated outputs, and deploying models responsibly with clear guidelines and oversight.

Imagine an AI that writes stories. It might accidentally create stories that are unfair or hurtful to certain groups of people because the information it learned from was biased. To prevent this:

- **We need to carefully test the AI** to see if it creates any biased or harmful content.
- **We should have systems in place to filter out** any inappropriate or dangerous outputs.
- **Finally, we need clear rules and someone watching over how the AI is used** to make sure it's used responsibly and doesn't cause any harm.

Q46: How do you train a generative model effectively with limited or noisy data?

Answer: To train generative models with limited or noisy data, strategies like transfer learning, data augmentation, regularization, and self-supervised learning are employed. Transfer learning uses pre-trained models to improve performance on similar tasks, while data augmentation generates additional training samples by modifying existing data.

Regularization techniques such as dropout prevent overfitting to noisy data and self-supervised learning enables models to learn patterns from unlabeled data, making them more effective.

Q47: Explain the importance of a Validation Set in generative AI.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Answer: A validation set guides hyperparameter selection, model architecture decisions and early stopping. While generative tasks are subjective, validation metrics and human evaluation on validation sets ensure the chosen model generalizes and doesn't overfit the training distribution.

Q48: Summarize current trends in Generative AI.

Answer: Key trends:

- Large multimodal generative models (e.g., text-to-image/video) with controllable generation.
- Diffusion models surpassing GANs in image quality and stability.
- LLMs integrated with retrieval and instruction-following, producing more truthful and user-aligned outputs.
- Tools like RLHF, watermarking, and content filters address ethical, safety, and IP concerns.
- Rapid advances in model compression, efficient sampling, and real-time generation empower broader deployment.

Q49. Can you describe a challenging project involving generative models that you've tackled?

Answer: One challenging project I've tackled involved generating realistic and diverse human faces using a Generative Adversarial Network (GAN). The project required careful consideration of factors like facial symmetry, realistic skin textures, and diverse ethnic features. Overcoming the instability of GAN training and ensuring the generated faces were free from biases was a significant challenge.

You can update this answer as per your own challenge as you learn and grow.

Q50.What are the hardware requirements for training large Generative AI models?

Answer:

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

significantly speed up the process.

2. Large memory capacity is crucial for storing model parameters, especially in the case of large Generative AI models.
3. Fast storage solutions, such as Solid State Drives (SSDs), are used to enable quick data retrieval and storage during training.
4. Distributed computing clusters with multiple GPUs are employed for parallel processing, reducing training time.
5. Access to high-speed internet is necessary for downloading and transferring large datasets, as well as for accessing cloud-based resources for training.

[Comment](#)[More info](#)[Advertise with us](#)

Next Article

Bayesian Optimization in Machine Learning

Similar Reads

Top Artificial Intelligence(AI) Interview Questions and Answers

As Artificial Intelligence (AI) continues to expand and evolve, the demand for professionals skilled in AI concepts, techniques, and tools has surged. Whether preparing for an interview or refreshing your knowledg...

15+ min read

Celebal Technology Interview Experience

Celebal Innovations Interview Insight (Nearby) 2023. Cerebral Advances visited our school to select freshers for the job of Partner computer programmer. The entire enrollment process had four rounds. One of which...

3 min read

Top Challenges for Artificial Intelligence in 2025

In 2025, Artificial Intelligence (AI) will lead the way in innovation and bring big changes to different fields. But with all the exciting possibilities, AI also faces some tough problems. Be they technical challenges or ethical...

6 min read

Tharsio Limited Interview Experience | Machine Learning Internship

I had a great experience during my interview for the Machine Learning Internship at Tharsio Limited. There

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Quantiphi Interview Experience for Machine learning engineer 2024 (On-Campus)

Recruitment ProcessRound 1 -> MCQ + codingRound 2 -> CodingTechnical round 1Technical round 2Technical round 3HR roundRound 1It consisted of 78 MCQ based on OS, CN, DBMS, JS, HTML, Probability,...

2 min read

Accenture Interview Experience For AI Engineer

As AI continues to transform industries, the demand for skilled AI engineers has skyrocketed. Recently, I had the opportunity to interview for an AI engineer position at an Accenture company. In this article, I'll share my...

2 min read

Salesforce Interview Experience | Set 5 (On-Campus)

Salesforce came to our college to hire for Full time Software Engineer role. The process was open to only Computer Science students with an aggregate CGPA > 7.0. The entire interview process was conducted on...

3 min read

Microsoft IDC Bangalore Interview | Set 153 (O365 Team)

For experienced profile Round 1: General discussion on career so far. Difference between process and thread Given a long string and a small pattern, find whether any permutation of the pattern is present in the string...

2 min read

FactSet Interview Experience | (On-Campus)

Factset visited our college in August, 2018 and allowed "C.S.E", "IT", "ECE" to sit in their recruitment test. Round 1: The first round was a pen-paper coding round. It consisted of three questions. Whole code was not...

3 min read

Expedia Interview Experience for ASDE Summer Internship

Round 1-Coding Round: The coding round comprised of two questions: Implement search in a BST - <https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion>. Given a list of Names, find th...

3 min read



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305


[Advertise with us](#)
Company

- [About Us](#)
- [Legal](#)
- [Privacy Policy](#)
- [In Media](#)
- [Contact Us](#)
- [Advertise with us](#)
- [GFG Corporate Solution](#)
- [Placement Training Program](#)
- [GeeksforGeeks Community](#)

Languages

- [Python](#)
- [Java](#)
- [C++](#)
- [PHP](#)
- [GoLang](#)
- [SQL](#)
- [R Language](#)
- [Android Tutorial](#)
- [Tutorials Archive](#)

DSA

- [Data Structures](#)
- [Algorithms](#)
- [DSA for Beginners](#)
- [Basic DSA Problems](#)
- [DSA Roadmap](#)
- [Top 100 DSA Interview Problems](#)
- [DSA Roadmap by Sandeep Jain](#)
- [All Cheat Sheets](#)

Data Science & ML

- [Data Science With Python](#)
- [Data Science For Beginner](#)
- [Machine Learning](#)
- [ML Maths](#)
- [Data Visualisation](#)
- [Pandas](#)
- [NumPy](#)
- [NLP](#)
- [Deep Learning](#)

Web Technologies

- [HTML](#)
- [CSS](#)
- [JavaScript](#)
- [TypeScript](#)
- [ReactJS](#)
- [NextJS](#)
- [Bootstrap](#)
- [Web Design](#)

Python Tutorial

- [Python Programming Examples](#)
- [Python Projects](#)
- [Python Tkinter](#)
- [Web Scraping](#)
- [OpenCV Tutorial](#)
- [Python Interview Question](#)
- [Django](#)

Computer Science

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

DevOps

Database Management System	AWS
Software Engineering	Docker
Digital Logic Design	Kubernetes
Engineering Maths	Azure
Software Development	GCP
Software Testing	DevOps Roadmap

System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Interview Preparation

- Competitive Programming
- Top DS or Algo for CP
- Company-Wise Recruitment Process
- Company-Wise Preparation
- Aptitude Preparation
- Puzzles

School Subjects

- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar
- Commerce
- World GK

GeeksforGeeks Videos

- DSA
- Python
- Java
- C++
- Web Development
- Data Science
- CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).