

Adobe

Adobe Inc. previously known as Adobe Systems Incorporated was founded by John Warnock and Charles Geschke in December 1982, a global computer software corporation based in the United States.

Whether or not you are involved in a creative career, you may have used Adobe software previously. Have you ever downloaded or generated a PDF document? In the 1990s, Adobe created the PDF file format. Adobe's flagship product, Photoshop, has become a verb, similar to Google, with a mission to 'promote creativity for all'. Aside from creating design and creativity software that is utilized by graphic designers, filmmakers, publishers, and students all around the world, Adobe is a great place to work.

The San Jose-based firm was named one of the 'Best Workplaces for Millennials 2020' by Fortune Magazine, and it was ranked #2 in the 'World's Most Admired Companies in Software'. After the company hit a revenue record despite the epidemic, CEO Shantanu Narayen, who has a 98 per cent approval rating on Glassdoor, revealed plans for a hiring drive in Q4 2020. Aside from high employee happiness, Adobe provides a variety of benefits, including onsite yoga and cafes, paid family holidays, and professional development services like LinkedIn Learning and Harvard ManageMentor on demand.

The corporation employs more than 22,000 people in 70 offices across 26 countries. Adobe recruits nine levels of software developers, including level I, with an average income of \$143,664. Adobe also has a university program for current students, and each year it hires over 1,000 interns and new graduates.

Adobe Recruitment Process

1. Interview Process

1. For Freshers:

Freshers are hired by Adobe through campus placement (Tier 1 colleges) and off-campus recruitment. Here are the main steps in the Adobe fresher recruitment process.

- Visit campus or apply online through a career site or a recruitment agency.
- Shortlisting CVs
- There is a written test for shortlisted ones. This includes writing pseudocode and, on occasion, answering essay-style questions.
- Interviews with experts (multiple rounds)
- Interviews with Human Resources

2. For Experienced:

In its Noida and Bangalore headquarters in India as well as its global offices, Adobe hires skilled professionals for various jobs in the technology and product development team. Here is the procedure for hiring experienced people at Adobe:

- Applying through a career portal or a staffing firm
- Shortlisting of CVs for an Adobe position
- Discussion over the phone (pre-technical round)
- Coding assignment on the Internet
- Adobe conducts technical interviews (multiple rounds like coding round, system design round, managerial round)
- Round of Human Resources

3. CV shortlisting procedure:

During the CV screening process, Adobe has stringent requirements. Here are some of the things they look for during the CV screening process.

- In India, Adobe selects applicants who have a grade point average of at least 70% or a CGPA of at least 7.0 throughout their studies (Class 10, 12, Graduation).
- Applicants should not have any educational backlogs.
- Candidates with a solid background in computer science, problem-solving skills, and a desire to code are preferred. In their work or projects, candidates should have proved their technical competence.

4. Online coding round at Adobe:

Candidates must exhibit their technical ability in the Online coding round after passing the CV screening round (for off-campus placement and for experienced professionals). Adobe uses popular sites such as Hackerrank, CodeinGame, and others to conduct online coding tests. During a recent Adobe recruitment, a hackerrank question was asked. The candidate must programme the process and obtain the desired outcome in this case.

2. Interview Rounds

1. Phone Screenings: The purpose of a normal phone screen is to familiarise the candidate with the firm and the open position, as well as measure their interest. The recruiter will inquire about your previous work experience and topic knowledge.

2. A phone interview with a hiring manager: If you pass the phone screening, you'll have a phone interview for the first round. A recruiting manager will go over your résumé in further detail, evaluating your leadership abilities, problem-solving style, and ability to work as part of a team. Prepare to talk about previous projects you've worked on and describe the process and results.

3. Technical Assessment: Successful candidates will be sent a link to an online technical assessment with up to 65 questions divided into two sections:

- Aptitude and logic (45 questions, with 45 minutes to complete)
- Technical and Coding (15-20 questions, with 75-120 minutes to complete)

With IQ-style questions, the aptitude section examines your quantitative and logic-based reasoning. Arithmetic algebra, profit-and-loss computations, and percentages are examples of quantitative problems, while riddles and data interpretation are examples of logic issues. Coding related questions are also tested. The technical coding portion of the test will be conducted on the HackerRank platform. To prepare, consider completing a couple of the site's example code problems ahead of time. Expect to be challenged on data structures, algorithms, and bit manipulation. C, C++, and Java are Adobe's preferred programming languages, but you can take the test in any language you want.

4. Onsite Interview: Because the whole worldwide workforce is working from home during the pandemic, Adobe's final-round interviews are being conducted remotely. A typical onsite includes lodging, daily meal reimbursement, and transportation to headquarters. These interviews are typically lengthier than off-site interviews. Expect 6-8 hours of back-to-back sessions lasting around 45 minutes each. There will be four rounds of technical interviews and one final HR round. Each round is an elimination round; if you are dismissed early, you will not receive an offer.

There are two rounds of coding.

- Interview for system design
- Interview about object-oriented design

Because the interviews will primarily consist of whiteboard coding, be prepared to describe your thought processes, such as why you chose a specific programming language, the alternatives forgone due to constraints, and the problem's outcome.

5. HR Round: Behavioral and situational interview questions make up the last HR phase. While Adobe places a premium on cultural fit, behavioural inquiries are often saved for the last round. They'll want to learn more about you, such as what you value and what kind of employee you are.

Adobe Interview Preparation

1. Interview Preparation Tips

1. Before the interview:

- Update your résumé and, in particular, your LinkedIn profile; if possible, include deliverables and metrics as real examples of your accomplishments.

- Remember that everything you put on your CV might be used against you, so be sure you know what you're doing.
- Spend at least two minutes talking about each point on your resume and mapping your accomplishments and past experiences to their fundamental values: genuine, extraordinary, innovative, and involved, as a good practice.

2. For the Interview:

- It's advisable not to try to memorise certain questions, in our opinion. There are no silver bullets in this world.
- Because firms of this size are continually striving to stay ahead of the curve and try new things, the questions they ask are constantly changing. The types of questions you'll be asked will vary depending on your team and the hiring manager.
- Instead, work your way through the foundations so you can grasp the underlying principles and confidently respond to even new types of interview questions.

Frequently Asked Questions

1. Is it difficult to get a job at Adobe?

There are both difficult and easy questions throughout the interview. Adobe's interview method differs little from that of other organisations. Even when hiring seasoned engineers, they conduct written tests. Overall, Adobe's approach is one of the most difficult.

2. What is Adobe assessment

The written assessment has a total time limit of 120 minutes. The Verbal Ability, Quantitative Aptitude, and Analytical Reasoning parts of the Aptitude Test are divided into three divisions.

3. Does Adobe recruit in India?

Adobe is now employing a large number of freshers as well as 2019, 2020, 2021, and 2022 graduates across India, mostly in Hyderabad, Chennai, Bangalore, Mysore, Cochin, Maharashtra, Mumbai, Delhi, and Coimbatore.

4. Does Adobe hire off campus?

Yes, Adobe does hire off-campus. The following steps should be followed:

- Visit Adobe's official website at adobe.com.
- Select Adobe Careers from the drop-down menu.
- Below you will see some blank information.
- Full-time/part-time, job category, location, and job type
- Then press the Submit button.

5. How many rounds of interviews does Adobe have?

There will be four rounds of technical interviews and one final HR round. Each round is an elimination round; if you're dismissed early, you'll receive no offers.

6. What is the culture of Adobe?

Adobe's business culture has won numerous awards. Adobe invests in its employees and the community with a strong sense of corporate responsibility, in addition to substantial employee incentives. Businesses may recall what's important by defining fundamental company principles.

7. How long will the interview process take?

It generally takes 1.5 months to complete the entire interview process.

8. What's the average salary for a software engineer?

The expected average salary an engineer can expect is \$110000 per annum.

Adobe Technical Interview Questions for Freshers

1. What is the use of Breakpoints in Adobe?

Breakpoints in Adobe represent different browser widths in pixels in a responsive layout. They are used to improvise as well as visualize designs in different browser widths on any platform, whether it is tablet, mobile, or desktop. Also, we can use breakpoints to test changes in the objects of a page with respect to the changes in browser widths.

2. What are shorthand properties, and mention their use?

Background, border, font, outline, padding, margin, and list are known as the shorthand properties. We use shorthand properties to write multiple properties in a single line. As a result, we can create an efficient block of codes and decrease lines of code significantly.

3. What do you mean by the Function Prototype?

The function prototype essentially declares in terms of the function's name, return type, and parameters. The compiler calls the function prototype when it is required in a program.

The syntax of the function prototype is given as follows

```
return_type function_name (parameter_list);
```

4. Brief about the Void Keyword in C++.

- The void keyword denotes that a function doesn't return a value if we use the keyword as a function return type.
- Similarly, the void keyword indicates that it takes no parameters if we use the keyword as a function's parameter list.
- Also, the void keyword denotes that a pointer is universal if we use the keyword for declaring a pointer.
- In short, the void keyword doesn't return a value or parameter, and we use the keyword as a universal pointer.

5. What do you mean by the Default function C++?

The function can be used as a parameter when a calling statement doesn't pass arguments. Generally, the default functions are assigned in the parameter list. Here, the compiler sets the parameters. Note that the Default functions are also known as the default arguments.

Syntax

```
Void funcName (type param value {  
}  
// code block
```

6. What is the need for memory protection in Operating Systems?

We can protect memory in operating systems with the help of the relocation register as well as the limit register. When it comes to the relocation register, the register has the smallest physical address, whereas the logical register has a range of logical addresses.

When an OS is protected correctly, it prevents processes from accessing unallocated memory spaces. So you can protect operating systems from harmful, malicious attacks.

We can protect the OS memory in many ways, such as memory protection using keys, masks, rings, dynamic tainting, simulated segmentation, and also capability-based addressing.

7. What do you mean by the Conditional Operator in C?

The conditional operator is also called the ternary operator. Mainly, it is used to evaluate Boolean expressions. It is also the decision-making statement. Besides, the conditional operator is an alternative to the 'If...else' statement. Every conditional operator has three operands.

The conditional operator's syntax is as follows

```
Expression1? expression2: expression3;
```

8. What do you understand about the exit-controlled loop?

In this loop type, the condition is executed only after executing the codes in the loop body. The codes are executed at least once for exit controlled loop. Note that 'Do.....while' is one of the examples of exit-controlled loops.

9. Define Shallow copy and Deep copy in Python.

A shallow copy in Python creates a new object and then populates it with references to child objects. Note that shallow copying doesn't recurse, so it doesn't create copies of child objects.

A deep copy also creates a new object but adds copies of the child objects recursively. It creates an independent clone of the original object and its children.

10. Explain Canny's algorithm.

We can detect edges in images using this algorithm. It consists of five steps as follows

- **Reducing Noise:** Noises in images are removed by applying the image convolution technique along with a Gaussian Kernel.
- **Calculating Gradient:** In this step, edge intensity, as well as the direction in images, are detected with the help of edge detection operators.
- **Suppressing Non-maximum:** We perform this step to make the edges of the images thinner
- **Double Threshold:** We achieve this step by identifying pixels of various types, such as weak, strong, and non-relevant.
- **Tracking Edge by Hysteresis:** In this step, we transform weak pixels into strong pixels.

Adobe Technical Interview Questions for Experienced

11. Write a code to find all the palindromes in a given string using C.

```
#include <iostream>
#include <string>
#include <unordered_set>
using namespace std;
// Expand in both directions of low and high to find all palindromes
void expand(string str, int low, int high, auto &set)
{
// run till `str[low.high]` is not a palindrome
while (low >= 0 && high < str.length() && str[low]
{
}
}
// push all palindromes into a set
set.insert(str.substr(low, high
// Expand in both directions
low--, high++;
-
low + 1));
==
str[high])
// Function to find all unique palindromic substrings of a given string
void findPalindromic Substrings (string str)
{
// create an empty set to store all unique palindromic substrings
unordered_set<string> set;
for (int i = 0; i < str.length(); i++)
{
}
// find all odd length palindrome with `str[i]` as a midpoint
expand (str, i, i, set);
// find all even length palindrome with `str[i]` and `str[i+1]` as
// its midpoints
expand (str, i, i + 1, set);
```



```

// print all unique palindromic substrings
for (auto i: set) {
}
}
cout << i << " ";
int main()
{
string str = "microsoft";
findPalindromic Substrings (str);
return 0;

```

Output

```
t foso im scro
```

12. State the differences between #define and Typedef in C language.

#define	Typedef
It is a directive	It is a keyword
It is used to determine a constant value for existing values	It is used to create a new name for the existing data types
It doesn't follow scope rules	It follows the scope rule
Preprocessor performs define statements	The compiler usually performs typedef interpretation
It shouldn't be terminated with a semicolon.	It should be terminated with a semicolon.

13. Write a code to swap two integers in C language.

```

#include<stdio.h>
int main() {
}
double first, second, temp;
printf("Enter first number: ");
scanf("%lf", &first);

```

```

printf("Enter second number: ");
scanf("%lf", &second);
// value of first is assigned to temp
temp = first;
// value of second is assigned to first
first
second;
// value of temp (initial value of first) is assigned to second
second = temp;
// %.21f displays number up to 2 decimal points
printf("\nAfter swapping, first number = %.21f\n", first);
printf("After swapping, second number = %.21f", second);
return 0;

```

14. Differentiate New () Vs. Malloc ()

New ()	Malloc ()
It calls constructors	It doesn't call constructors
It is an operator	It is a function
It returns a data type	It returns void*
It throws bad_alloc exception on failures	It returns a NULL value on failures
Memory is allocated from free store	Memory is given from the heap
It doesn't change the size of the buffers	It changes the size of buffers

15. Write a code in C++ to count the complete set of bits in an integer.

```

#include<iostream>
using namespace std;
//Count total set bits in a number
unsigned int bits (unsigned int number){
unsigned int count = 0;
unsigned i;

```

```

}
//display the total 8-bit number
cout<<"8-bit digits of "<<number<<" is: ";
for (i = 1 << 7; i > 0; i = i / 2){
}
(number & i)? cout<<"1": cout<<"0";
//calculate the total set bits in a number
while (number){
}
count += number & 1;
number >>= 1;
cout<<"\nCount of total set bits in a number are: "<<count;
int main(){
}
int number = 10;
bits (number);
return 0;
8-bit digits of 10 is: 00001010
Count of total set bits in a number are: 2

```

16. Compare Call by Reference and Call by Value.

Call by Reference

They are functions that occur when we pass addresses for variables

We can change the actual values of variables using function calls

We need pointers to pass variables

We can access the actual values of variables and change them

Call by Value

They are functions that happen when we pass values for variables.

We cannot change the actual values of variables using function calls

We can simply pass variables

Changes made in dummy variables do not affect the actual values of variables.

17. Write a C program to find the pair of numbers whose addition equals the given number.

```

#include <stdio.h>
#define ARRAY_SIZE 100000
void hasSumPair(int array[], int size, int sum) {
int i;
/* NOTE : here we are assuming that all numbers
in input array are less than 100000 */
int table [ARRAY_SIZE] = {0};
for (i = 0; i < size; i++) {
if(table[sum-array[i]] == 1 && sum-array[i] >= 0) {
printf("Found Pair : %d %d\n", array[i], sum-array[i]);
table[array[i]]
}
}
}
=
1;
int main(){
int i, array[1000], count, sum;
printf("Enter the number of elements in Array\n");
scanf("%d", &count);
printf("Enter %d numbers\n", count);
for(i = 0; i < count; i++){
}
scanf("%d", &array[i]);
printf("Enter the value of sum\n");
scanf("%d", &sum);
hasSumPair(array, count, sum);
return 0;
}

```

Output

```

Enter the number of elements in Array
6
Enter 6 numbers
7 2 4 3 15
Enter the value of sum

```

18. What do you mean by Stack overflow error, and how to avoid it?

Stack overflow error is nothing but a type of buffer overflow and logical runtime error. This error occurs when the allocated stack memory is insufficient to execute a program. Mainly, recursive functions cause a stack overflow. It also occurs because of the programming language, the host computer architecture, and available memory.

To avoid the [stack overflow error](#), we need to define the call stack size at the start of a program.

19. Write a code to find the height of a binary tree.

```
#include <stdio.h>
#define ARRAY_SIZE 100000
void hasSumPair(int array[], int size, int sum) {
    int i;
    /* NOTE : here we are assuming that all numbers
    in input array are less than 100000 */
    int table [ARRAY_SIZE] = {0};
    for (i = 0; i < size; i++) {
        if(table[sum-array[i]] == 1 && sum-array[i] >= 0) {
            printf("Found Pair : %d %d\n", array[i], sum-array[i]);
            table[array[i]]
        }
    }
}
=
1;
int main(){
    int i, array[1000], count, sum;
    printf("Enter the number of elements in Array\n");
    scanf("%d", &count);
    printf("Enter %d numbers\n", count);
    for(i = 0; i < count; i++){
    }
    scanf("%d", &array[i]);
    printf("Enter the value of sum\n");
```

Overloading

It follows compile-time polymorphism

It only occurs between the methods of the same class

We can view errors only at compile time

Methods must have different signatures

Static binding is applied in overloading

Overriding

It follows runtime polymorphism

It occurs between the superclass and subclass

We can view errors only at the run time

Methods must have the same signatures

Dynamic binding is used in overriding

```
scanf("%d", &sum);  
hasSumPair(array, count, sum);  
return 0;  
}
```

20. State the differences between Overloading and Overriding.

Adobe Leadership Principles

Leaders in Adobe are always keen on demonstrating high-value principles to create a positive and conducive work environment.

Let's have a take on Adobe leadership principles as follows.

- *Genuine*

Leaders in Adobe are always being true to their employees and customers. They play a pivotal role in tending their workforce to be creative and productive. They demonstrate it in multiple channels. Mainly they encourage diversity among their employees. They treat people equally and respectfully regardless of race, gender, country, religion, ethnicity, etc.

- *Exceptional Experiences*

The leaders of Adobe are keen on creating unique experiences for their employees and customers. They want to delight people connected to Adobe in one way or another.

- *Highly Creative*

Adobe leaders are highly creative and interested in innovation. Not just being creative, they implement the potential ideas into the business. In simple words, they convert ideas into business outcomes.

- *Involved*

Adobe leaders sincerely involve all the people connected to their business – who might be customers, partners, employees, or communities. In other words, they practice intense, open, and active engagement with Adobe-associated people.

- *Encourage Learning*

This is yet another leadership principle that Adobe leaders follow. Adobe encourages continuous learning for its employees by offering attractive learning programs. The learning programs will support the employees in their personal growth as well as career development in a significant way.

Tips to Crack Adobe Interview

Are you interested in knowing the tips to crack Adobe interviews? Follow the below pointers to nail your Adobe interview effortlessly.

#Tip 1 - Enhance Technical Expertise

You must prepare key computer science domains such as C, Python, C++, data structures and algorithms, databases, basics of data science, operating systems, system design, and more. By doing so, you can attend the online test and technical interviews confidently. You must be clearly brief about your previous projects if you are an experienced candidate. If you are a fresher, you must brief about what you have learned from your interns and certifications.

#Tip 2 - Practice Coding

Before attending your Adobe interview, be thorough in writing algorithms and coding. Mainly, you must be thorough in algorithms such as binary search, breadth-first search, depth-first search, merge sort, divide and conquer, quick sort, and others. Multiple coding platforms are available online, such as TopCoder, HackerRank, Codility, LeetCode, CodeChef, etc.

#Tip 3 - Present effectively

Presentation is highly-essential rather than your preparation to deliver the best. If you don't present effectively to interviewers, your practice and hard work will be useless. Be clear with your approach. You must precisely convey what you want to say consistently. You can practice mock interviews with your family, friends, or colleagues to improve your presentation skills.

#Tip 4 - Answer tactfully

In an Adobe interview, interviewers could ask questions based on your experiences, challenges, and so on to test your attitude and behavior. Also, you must answer interviewers in such a way that you have solved challenges by applying your analytical skills, taking team support, and employing innovation. To answer these types of questions, you can follow the STAR format. STAR refers to Situation, Action, Task, as well as Result.

What's more! You must have self-confidence and courage as the fundamental qualities to nail an Adobe interview.