

# Whatfix Placement Drive



Date : 11<sup>th</sup> October, 2018

## About

Whatfix enhances self-service capability of web products and helps companies to reduce support queries and improve user engagement. It is a cloud platform using which product teams can self-curate interactive guides & integrate across all user touchpoints. If you've ever seen an enterprise product which is complex and you don't know how to go about using it and make maximum out of it, then we fix that with our interactive guides. Our product is built around a core of JAVA with enterprise businesses using the Whatfix system. On a monthly basis, millions of interactive guides are processed. Building, testing, and deploying to any of our environments are as simple as pushing a commit to a git branch. We like to keep our technology up-to-date. All of the technology is built and maintained by this small development team, so from day 1, software you write will be used by people from all over the world.

## Job Description

**Profile :** Full Stack Engineer ( SLI + FTE )

**Roles and Responsibilities :**

What are we looking for?

We are looking to bring in experienced Full Stack Engineers to make our engineering core stronger. We're looking for professional full stack development experience, which means not only should you have JAVA experience; you should also be hands on with front end technologies. If this sounds interesting, we would love to hear from you.

What you will do?

- Take ownership of the sub-system.
- To understand the customer needs, design & implement.
- Occasionally participate in customer demos & provide inputs wherever required.
- Would be responsible for anything from front-end to back-end activities.
- Keep optimizing & automating mundane tasks.

What you should must have:

- Hands on experience on Java, JS, Cassandra
- Strong verbal and written communication skills
- Work experience, preferably from internet companies, products or startups.

## Eligibility

CGPA  $\geq$  5 (no active backlog)

Branch : CSE | CCE | ECE

## Procedure

1. Online Test ( Apti + Coding) - 60 min
2. Technical Interview (3)
3. HR Interview

## Result

No. of Selections : 1

Candidate : **Pushkar Garg**



# Interview Experience

**By :- Pushkar Garg**

The process consisted of 5 rounds.

## **Round 1: Online screening test(Coding+Aptitude)**

-Duration-1 hour

-Language-JAVA

-It consisted of 10 MCQs(based on JAVA+HTML) and 2 coding questions(fairly easy just required some logic)

## **Round 2: Personal Interview**

Questions based on JAVA, OOPs, time and space complexity were asked. Need to be aware with the hierarchy of Java library and their time complexities in detail to clear this round.

## **Round 3: Personal Interview**

Here my data structures and algorithms and my problem-solving skills were judged. The interview began with the hierarchy of map interface in Java and the time complexities. Then questions based on tree data structure were asked mainly AVL tree and red-black tree. Then 2 coding questions were given. First was Kadane's algorithm and the other was to check if the input number is palindromic or not.

## **Round 4: Personal Interview**

In this round, 3 questions were given and I was asked to code them and tell time and space complexities. Fairly easy questions requiring a bit of logic and knowledge of Collections framework in Java.

## **Round 4: HR Round(telephonic)**

Just my future goals were asked. Just to see if I'm fit for the company.

# Question Bank

## **Round 1**

1. Stack v/s queue
2. Implementation of stack using array and linked list
3. Inheritance examples
4. Encapsulation examples
5. Abstraction examples
6. Give an algorithm to find max number formed by number given in an array.
7. HashMap and HashSet working
8. Time complexities of Linked hash set, tree set, hashset
9. Implementation of hash/tree set
10. Call by reference and call by value
11. Dealing with infinite array, searching through stream of infinite integers
12. Overloading, overriding methods
13. Interfaces and extending classes
14. Polymorphism
15. Hierarchy of Java classes
16. String concatenation in system.out
17. Exception handling
18. Time complexity of library methods like set, list, map
19. Swap two variables without using 3rd variable
20. Occurrences of each letter and print in nondecreasing order of occurrences



21. Find max and min numbers from one billion numbers and time complexity for this
22. What are generics
23. Difference between set and list
24. Difference between interface and abstraction
25. Stack: push, pop, their complexities
26. How searching works in set
27. What are objects
28. When and how static keyword works?
29. What is inheritance and runtime polymorphism?
30. Find max() in stack.
31. Multithreading
32. Mysql
33. There are N cities and ith city has population A[i]. There are k hospitals. Distribute hospitals in such a way that there are minimum number of hospitals per patient (GREEDY).
34. BST, sorting
35. Constructor overloading
36. Priority queue
37. Print ith element of the pattern : 112123123412345.....
38. Access modifiers
39. Print element in spiral form in an array
40. With N distinct objects we need a class which has to have add, delete, search and sort. Which D.S. should be used to store the data.
41. Difference between process and threads.
42. Given histogram : find the max and min area that can be calculated without cutting edge. (Dynamic Programming)
43. How a java server works?
44. Distributed system.
45. Find duplicate words in a string.
46. Given a delete method which iterates over the list and deletes it from the list. Which would you prefer array or linked list and why?
47. Given an array having knowledge of students. Students need to divide themselves in pair such that difference between best project team and worst project team is minimum. Knowledge is between 2 and 50. For a pair knowledge is sum of knowledge of both students in the group.

## Round 2

1. Frequency of elements using hashmap.
2. Number of keys in hashmap.
3. Difference between list and a set.
4. Searching time complexity in hashmap.
5. Hierarchy of map interface in java library and time complexity of classes that implement this interface (methods in classes).
6. Different types of trees : AVL, BST, Red black
7. Kadane's algorithm, find if input number is a palindrome or not.

## Round 3

1. Union and intersection of 2 arrays.
2. Sort the array of 0's and 1's.
3. Merge, insertion sort
4. Why insertion, deletion time complexity of hashset is O(1).