How I cracked TCS Digital

Hurrah! My TCS Digital interview results are out, and I thankfully made it. Having wanted to work in the field of Artificial Intelligence for a long time now, I don’t think there is a better platform than TCS Digital for learning how to do so in the professional world, for freshers like us.

In this blog, I’ll share my preparation strategy, the resources I used, and even my codebase for going through all the stages, and cracking them one by one.

**Intended audience.** This article is targeted toward students pursuing engineering, and are currently in their final year (irrespective of their stream). However, if you are in one of your earlier years, this article will still help you can get early start, which in fact is a big advantage.

**About Digital Technologies**

Firstly, you should know what these technologies are, as you will be working with them if you get selected. Knowing about these technologies will help you get through the later stages of the recruitment procedure.

Key terms you should know about (feel free to check other resources to know about them):

* [Artificial Intelligence](https://courses.edx.org/courses/BerkeleyX/CS188.1x-4/1T2015/course/)
* [Machine Learning](https://www.coursera.org/learn/machine-learning)
* [Deep Learning and NLP](https://www.coursera.org/specializations/deep-learning)
* [Cloud Computing](https://en.wikipedia.org/wiki/Cloud_computing)
* [Computer Vision](http://www.bmva.org/visionoverview)
* [Blockchain](https://www.blockchain.com/)
* [Virtual Reality](https://en.wikipedia.org/wiki/Virtual_reality)

It is not required to know about all these technologies in-and-out; I certainly didn’t. However, you should have *some* idea about all of them, and experience in working with at least *one*of them; and by experience, I mean projects.

Finally, you can read what TCS has to say about these technologies in their own website, [**TCS Interactive**](https://www.tcs.com/tcs-interactive).

**Stages in the Recruitment Process**

There are more than one way to get into TCS Digital; I will be mentioning the steps I took in the process, and go through each of them in detail.

Stages in the process:

1. CodeVita
2. Interview for Ninja
3. Online Test for Digital
4. Interview for Digital

These are the steps I followed. You might have noticed that these are the steps of an off-campus drive. If you happen to be a student of one of the premium colleges where TCS conducts its Digital recruitment, only follow steps #3 and #4.

For everyone else, start from the top,  *i.e.,*step #1.

Now, I unfold each of these phases one by one.

**Stage #1: CodeVita**

TCS hosts its own coding contest every year, called [CodeVita](https://campuscommune.tcs.com/en-in/intro/contests/codevita-2018" \t "_blank). It consists of algorithmic coding challenges, like most standard coding contests.

If you are a final year student and manage to secure a “good” rank in this contest, you will directly qualify for the interview, without having to sit for any aptitude test (which is why I prefer this step). If not, you will have to sit for NQT, and crack that test.

By a “good” rank, I mean ranks less than 3000 to be on the safe side, although you might still qualify if you are under 4500 in some cases. To be on the safe side, make sure you manage to solve at least 2 problems from the problem set.

Also, be thorough with the language you choose to solve your problem with; you might be asked questions from that very language in the interview. The preferred language for competitive coding is **C++**, although you may choose another programming language if it suits you.

**Resources for CodeVita.**

* [TCS CodeVita Past Questions](https://www.programminggeek.in/2016/07/TCS-CodeVita-Previous-years-questions.html)
* [The Complete Reference to Competitive Programming](https://www.hackerearth.com/getstarted-competitive-programming/)
* [CodeChef Practise](https://www.codechef.com/problems/school)
* [C++ for Programmers](https://in.udacity.com/course/c-for-programmers--ud210) (course)
* [Fork C++](https://practice.geeksforgeeks.org/courses/fork-cpp) (course on GeeksforGeeks)
* [C++ STL Tutorial](https://www.geeksforgeeks.org/cpp-stl-tutorial/) (on GeeksforGeeks)

**Stage #2: Interview for Ninja**

After you secure a good rank in CodeVita, you will probably be called for an interview. Try your best to crack this interview, as it will get you a “Generic Coders” job, so even if you don’t get Digital, you will still have a job.

The questions in this interview will vary, depending on your CV and the way you steer the discussion in the interview room.

**Important sections in CV.**

Your TCS-format CV will consist of different sections that will most likely govern the interview progress. Let me go through some of those in particular.

* **Trainings / Projects Undertaken.**Undoubtedly, this is one of the most important sections of your CV. Make sure you have at least two technical projects. Be thorough with them, as well as the technologies you used for them. For example, if you have mentioned a Banking System you wrote in Django, you should know how your project works, the overall structure of Django, Python programming, HTML / CSS, and JavaScript.
* **Subjects of Interest.**Needless to say, be well-versed with all the subjects you have mentioned. Do not simply list all the subjects you have taken so far; instead, mention only the ones you are most confident in. (Tips for Computer Science students: regardless of the subjects you mention, prepare OOP in Java, DBMS and practise SQL queries.)
* **Programming Languages.** Mention C and Java, and be through with both of them. You may also mention another object-oriented language in place of Java (like C++ or Python), if you’re more confident in them. [Click on this link for resources on these languages.](https://github.com/rafi007akhtar/CSE-resources#programming-languages)
* **Hobbies.**Believe it not, this section matters. Questions on hobbies are frequently asked, so mention only those hobbies you are truly passionate about.

Once again, cracking this interview would secure you a job in the “Generic Coders” category with a CTC of INR 3.36 LPA. To get a job in the Digital category, you need to go through the following two steps. If in case you don’t make it, you will still have your Ninja job.

**Stage #3: Online Test for Digital**

If you manage to crack the above interview, congratulations! This means two things:

1. You have secured a job in the “Generic Coders” category, aka TCS Ninja. Your performance in this stage and the next will not affect this job offer.
2. You are qualified to sit for TCS Digital, and here is where things get rough.

I don’t want to scare you, but I’m not going to lie either. This test is a *difficult*one. The good thing, though, is with ample practise, you can definitely crack it.

Firstly, let me mention the sections in this test. They are English / Verbal ability, Quantitative Aptitude, Lateral Thinking, Agility, Programming Logic, and Advanced Coding.

***You can get familiar with these sections, and practise them on the official mock link,***[***here***](https://g91.tcsion.com/OnlineAssessment/index.html?2030@@M1)***.***

As a summary, the following table shows the order and the time given for these sections.

| **SECTION NAME** | **ENGLISH** | **QUANTATIVE APTITUDE** | **LATERAL THINKING** | **AGILITY** | **PROGRAMMING LOGIC** | **ADVANCED CODING** |
| --- | --- | --- | --- | --- | --- | --- |
| **DURATION (IN MINUTES)** | 15 | 30 | 30 | 25 | 20 | 60 |
| **NO. OF QUESTIONS** | 10 | 12 | 7 | 2 | 7 | 1 or 2 |
| **ORDER OF APPEARANCE** | 1st group | 2nd group | 3rd group | 4th group | 5th group | 6th group |

**Notes.**

* The source of this table is the official mock link I shared above.
* Regarding the 6th group (Advanced Coding), the mock link has only *one*question, but we were given *two*questions to solve in the same time duration in the actual test. More on that below.

Let us now go through these sections, one by one.

**1. English**

This section will consist of *ten*questions ought to be solved in *fifteen*minutes. In the mock link, the questions draw from topics like verbs, prepositions, identifying the error in the sentence, identifying the correct sequence of sentence, and so on. It also contains questions of synonyms and antonyms of difficult words.

You check out the resources available in [Wordpundit](https://wordpandit.com/" \t "_blank) for improving your vocabulary.

**In the actual test, however, all the questions were passage comprehensions. Additionally, all the passages were about digital technologies.**This is why in the beginning of this article, I provided links to some of these technologies. Read through all of them, and get an insight about them so you don’t feel off-handed during the exam.

**Resources for English section.**

* [Top 10 Mindblowing Upcoming Technologies](https://www.wonderslist.com/top-10-mindblowing-upcoming-technologies/) [blog post to read about upcoming technologies]
* [IELTS Academic Test Preparation](https://www.edx.org/course/ielts-academic-test-preparation) [module 3 of this course deals with reading and comprehension]

**2. Quantitative Aptitude**

This section will contain the most challenging of aptitude questions: 12 questions to be solved within half-an-hour. The problems will be drawn from high-school mathematics, and will require a solid grasp of concepts and an ability to *think* for a solution in limited time. All of this can be nurtured through practise.

Currently, there are no official practise materials solely designed for the purpose of this section. What you can do it practise from whichever material you get your hands on, some of which I’ll be listing below.

**Resources for Quantitative Aptitude section**

* [Fast Track Objective Arithmetic](https://www.amazon.in/Track-Objective-Arithmetic-Rajesh-Verma/dp/9312149830/ref=sr_1_1?s=books&ie=UTF8&qid=1539518449&sr=1-1&keywords=arihant+objective+arithmetic) (book)
* [Quantitative Aptitude for Competitive Examinations](https://www.amazon.in/Quantitative-Aptitude-Competitive-Examinations-Aggarwal/dp/9352534026?tag=googinhydr18418-21&tag=googinkenshoo-21&ascsubtag=04a36ea3-bb19-49f1-80b3-eb2b424b0ecc&gclid=CjwKCAjw0oveBRAmEiwAzf6_rCBv4vOpCAEOfwMRTNKhV3OgY8RFvcQxs79qocgt4XYkJgFMTs6U6xoCOJMQAvD_BwE) (book)
* [Permutations and Combinations](https://www.youtube.com/playlist?list=PLEbnTDJUr_IeCndN2NGgj147ByWNSg1I1) (YouTube Playlist)
* [Practise Questions on GeeksforGeeks](https://www.geeksforgeeks.org/placements-gq/#GRP)

**3. Lateral Thinking**

*Monday, six friends went camping. Tuesday, John, Jack, and James cooked some mushrooms. Wednesday came and they ate the mushrooms. Thursday found them all dead. Exactly one friend survived, how come?  
Answer: Thursday is the name of the friend who did not have the mushrooms, and so he / she survived.*

You guessed it: this section will consist of *brainteasers.*

When you start this section, throw away any horizontal, vertical or even logical thinking, and get as lateral as you can! [Check out this YouTube video that clarifies the distinction between them.](https://www.youtube.com/watch?v=gB5ve_Bqf0M)

Be ready to solve rebus, prepare all possible words with meaning from a given word, find that missing link, identify the correct pattern?—?basically, any brainteaser you can think of.

**Go through the sample questions contained in the mock link**. The standard of questions in the real test will be pretty much the same. There will be lots of time (half-an-hour for 7 questions), so attempt *all*the questions.

**Resources for Lateral Thinking section**

* [Brainteasers](https://www.puzzleprime.com/brain-teasers/)
* [Lateral Thinking](https://www.amazon.in/Lateral-Thinking-Creativity-Step/dp/0060903252?tag=googinhydr18418-21&tag=googinkenshoo-21&ascsubtag=04a36ea3-bb19-49f1-80b3-eb2b424b0ecc) (book)

**4. Agility**

This section will contain *two*questions to be solved in 25 minutes time.

In the mock link, the first question is a simple API-based program in Java that needs to solved in the given editor. In this case, you need to read the question to know what is asks, and implement a program using the tools they provide you.

However, in the actual test, this question was replaced by a new language they created, that dealt with handling of the mathematical structure, [sets](https://en.wikipedia.org/wiki/Set_%28mathematics%29), and posed problems to solve in the said language. Naturally, you’d need to go through the language rules quickly, understand them, and solve the program in their editor.

The second question is an exhaustive comprehension on a mathematical concept / upcoming technology. In the mock link, the passage is on [Computer Vision](https://youtu.be/vT1JzLTH4G4), while in my test, it was on [Game Theory and Prisoner Dilemma](https://youtu.be/3Y1WpytiHKE). You were required to analyze the passage thoroughly, and draw inferences from it, on the applications of the said theory / technology.

Again, for the second question, you need to be aware of the latest and upcoming technologies in the digital world.

**Resource for Agility section**

* [How to use Java APIs](https://stackoverflow.com/questions/4504209/how-to-use-java-apis)

**5. Programming Logic**

If you’re a CSE student, you should already be familiar with the topics in this section.

*Recursion, arrays, stacks, queues, linked lists, trees, binary search trees, binary heaps, graphs.*

In short, you should know data structures and their implementations. Along with them, you should know standard algorithms and their space and time complexities. You will get 20 minutes to solve 7 questions.

**Resources for Programming Logic section**

* [Algorithms and Data Structures](https://www.youtube.com/playlist?list=PLEbnTDJUr_IeHYw_sfBOJ6gk5pie0yP-0) (YouTube playlist)
* [Practise Data Structures on GeeksforGeeks](https://www.geeksforgeeks.org/data-structures/)
* [Practise Algorithms on GeeksforGeeks](https://www.geeksforgeeks.org/fundamentals-of-algorithms/)
* [180+ Algorithm & Data Structure Problems using C++](https://github.com/mandliya/algorithms_and_data_structures) (GitHub repository)

**6. Advanced Coding**

As the name gives away, this section is an hour-long coding round.

In the mock link, this section consisted of a single question, but in the actual test, we were given *two*programs to solve in the same time constraint. However, those two programs were relatively easier than the one given in the mock link.

If you’re given a single question, it will probably be of the same standard as CodeVita Round 1 questions, so recheck the CodeVita section for this section.

**Languages allowed:**C, C++, Java, Python 2.7, Perl. (This set of languages may change overtime.)

**Additionally, you may have a look at**[**this Github repository**](https://github.com/rafi007akhtar/tcs-digital-prep)**.** It contains all the programs I solved in the two-weeks I had for this round.

**Integrated Development Environment.**Apart from the usual text-editor, we were also provided with [Eclipse Oxygen IDE](https://www.eclipse.org/oxygen/) to aid in our coding. I urge you to install the said IDE and familiarize yourself with how it works.

**Stage #4: Interview for Digital**

It might take a few days to assess your performance in the online test. Regardless, you should start preparation for this interview straightaway after the test.

**You will have a huge advantage going to the interview if you have projects made with the upcoming technologies**. For me, it was an image classifier written from scratch to recognize handwritten digits. (You can find that project [here](https://github.com/rafi007akhtar/Digit-Classifier).) The technology used was Deep Learning and Neural Networks.

Although I had 2 more projects of the same criteria, one of was a Sudoku Solver that used basic AI concepts to solve a Sudoku puzzle, and another was a cat-image recognizer, the interviewer insisted on talking about the above project.

Additionally, I had a few other projects under my belt to talk about in the interview. One of them was a [banking app](https://github.com/rafi007akhtar/CentralBank) written in Django, and we spent a considerable amount of time talking about it.

So here’s what I have to say: study the subjects you have mentioned in your “Subjects of Interest” section of the CV, but put your main focus on making and exploring these projects. Additionally, you should be able to explain every concept you used in bringing your project to actualization.

For example, if you are including any DL project, make sure you know important concepts like Gradient Descent, Neural Networks, Shallow Networks v/s Deep Networks, Activation Functions you used for different layers, Forward Propagation v/s Backward Propagation, and so on.

If you already have a bunch of projects of the above sort, you should be good to go. Choose the projects you intend to include wisely, and make sure you’d be able to explain everything about it in the interview room.

**Resources for Interview**

1. If you are getting started with Machine Learning, you check out [Udacity’s course](https://in.udacity.com/course/intro-to-machine-learning--ud120-india" \t "_blank) on the said topic. This course will not only explain the fundamental concepts needed but also provide hands-on mini-projects to apply your learning to actual datasets.
2. If you want to solidify your concepts on Machine Learning, I recommend the [course on Coursera](https://www.coursera.org/learn/machine-learning).
3. If you’re starting with Deep Learning, I highly recommend starting with Michael Nielson’s [online book](http://neuralnetworksanddeeplearning.com/). The first two chapters serve as an introduction to the topic, along with the image classifier project I was talking about earlier.
4. You can supplement your learning of DL by following [this series of DL courses](https://www.coursera.org/specializations/deep-learning) on Coursera as you read along the above book. Alternatively, if you are lacking in time, you can go through Udacity’s course on [Deep Learning](https://in.udacity.com/course/deep-learning--ud730-india).
5. Lastly, you can get started with AI by starting with [UC Berkley’s AI course](https://courses.edx.org/courses/BerkeleyX/CS188.1x-4/1T2015/course/)on edX.

**Note.**It is not required to complete all of these courses in the limited time you have. In my case, I had already done the first course in the series mentioned in point (4), so I revised all of that. Additionally, I completed the first two chapters of the book mentioned in point (2), along with the project given in it, and these two steered most of the interview. Apart from them, I had also done parts of the course mentioned in point (1), and parts of Udacity’s DL course mentioned in point (4), and started out with the AI course in point (5), all of which were extremely useful to know. So, you should divide your time accordingly, see which courses suit you the most that could be done in the given time, and do them.

**Ending notes**

Getting into TCS Digital isn’t easy, but it surely is rewarding. Study hard, study *smart,*and good luck!