

National Programming Skills Report

Introduction

It is our pleasure to present the first edition of Automata report. Programming Ability skill is the key for jobs in IT centric world.

In the last few years, thanks to jobs becoming global, the importance of good programmers has increased manifold. It has over the years become an important criteria of hiring for most of the IT software companies, both at the international and intra-national levels. A candidate with good programming skills is so important , because there are cases where one doesn't understand the program statement and if does then fails to implement it properly.

The report tries to identify patterns in employability across different regions, analyzing in detail the distribution of employability across various roles. Herein, it studies how employability varies across different groups: We do the first systematic study of Programming Ability of engineers across India. The study is based on Automata, an automated tool measuring programming skills used across the world by industries. Automata is a 60 minute test taken in compiler integrated environment. It rates the candidates on multiple parameters of Programming Ability , Programming Practices, run-time complexity and test case coverage. It uses advanced artificial intelligence technology to automatically grade programming skills.

The report follows the tradition of our National Employability Reports to uncover the gaps in Programming Ability by gender, tier of college, top 100 colleges vs. rest of the colleges, etc. Based on the reports, we have come up with specific suggestions on how to improve Programming Ability in various parts of the country.

With commitment to the development and progress of higher education in India!

Executive Summary

The key findings of the present study are as follows:

❑ Only 36% engineers are able to write compilable code

With the advancement in digital services and products, good programmers are the constant requirement of industry. Companies require candidates with strong understanding of computer programming and algorithm to implement on new innovative ideas in a time constraint workplace. Unfortunately, we find that out of the 2 problems given per candidate, only 14% engineers are able to write compilable codes for both and only 22% write compilable code for exactly one problem. To sustain the growth of IT industry, we need candidates with high technology caliber and better programming skills.

❑ Lack of adequate knowledge to build a logically correct, maintainable code is the key reason for low employability

Only 14.67% engineers are employable for IT Services company. While a worryingly low percentage of 2.47% are observed to be employable in IT Product company. It should be noted that this has been calculated according to the current requirement and hiring policy adopted by these 2 sectors. Where IT Services companies require trainable programmers with sound logics, who may or may not possess in-depth knowledge. However, IT Product companies require substantial competence in programming algorithms. The key reason behind such paltry employability percentages is deficient skills to apply basic programming and algorithm principles. These concept and principles are there in college curriculum, which indicates that there is a gap in implementation. Hence, these industries shall have a very hard time in hiring students for their organization who are able to fit in the criteria and yield required results.

Executive Summary

❑ **As low as 2.21% engineers possess the skill to write a fully functional code with best efficiency and writing practices**

Functionally correct code is the basic requisite of a good programmer. However, to enhance the quality of the code, few more important indicators have emerged - efficiency, time complexity and space complexity. Nothing is more time consuming than dealing with badly written code which leads to enormous bugs & exceptions. The analysis unveils that only 2.21% engineers possess the skillset to write logically correct code with best efficiency & least time-space complexity.

❑ **Programming Practices and Programming Ability are the areas of maximum skill gap across demographics**

Engineers show larger skill gap in Programming Practices and Programming Ability abilities. Programming Practices is the ability to code in readable & maintainable fashion and Programming Ability is the ability to code in most optimized way. We find that these two are the major areas of differences when compared across various demographics such as gender, tier of colleges, and top 100 colleges. There is a need to put emphasis on improving the code quality.

Table of Contents

- ❑ Introduction
- ❑ Executive Summary
- ❑ Methodology
- ❑ Tool Description : About Automata
- ❑ Programming Ability Score Distribution
- ❑ Distribution across demographics
 - ❑ By Gender
 - ❑ By Tier of college
 - ❑ By Top 100 campuses vs. the rest
 - ❑ By Zone
- ❑ Programming Ability across difficulty of problem

Methodology

This report is based on a sample of more than **36,800** students from **500+** colleges across India. The properties of the data set are given in the table below.

PARAMETER	STATISTICS
Gender ratio	Male-61.1%, Female-38.9%
% candidates in different Tiers of Colleges	Tier1-28.7%, Tier2-22.7%, Tier3-31.3%
% candidates in Top 100 colleges and rest of the colleges	Top 100 colleges - 15.3%, Rest of the colleges - 75.9%
% candidates in Key Cities (city of college)	New Delhi-3.0% Mumbai+Pune-6.5% Chennai-7.6% Bengaluru-3.8% Kolkata-1.7% Hyderabad-7.6%
% candidates in Zones (city of college)	East-10.2% North-26.7% West-14.3% South-47.5%

Together with the Automata scores, the various demographic details of the candidates are also captured by Aspiring Minds testing platform, which has enabled a comprehensive and meaningful analysis provided in the report.

Programming Skills

What is the key problem?

Automata is an **automated programming ability evaluation** tool that uses a simulation compiler integrated environment to predict a candidate's ability to develop code that is functionally correct, generates expected outputs, is efficient and follows best coding practices. A **curated question bank, validated empirically, with problems testing different aspects** of data structure usage and algorithmic approaches is used.

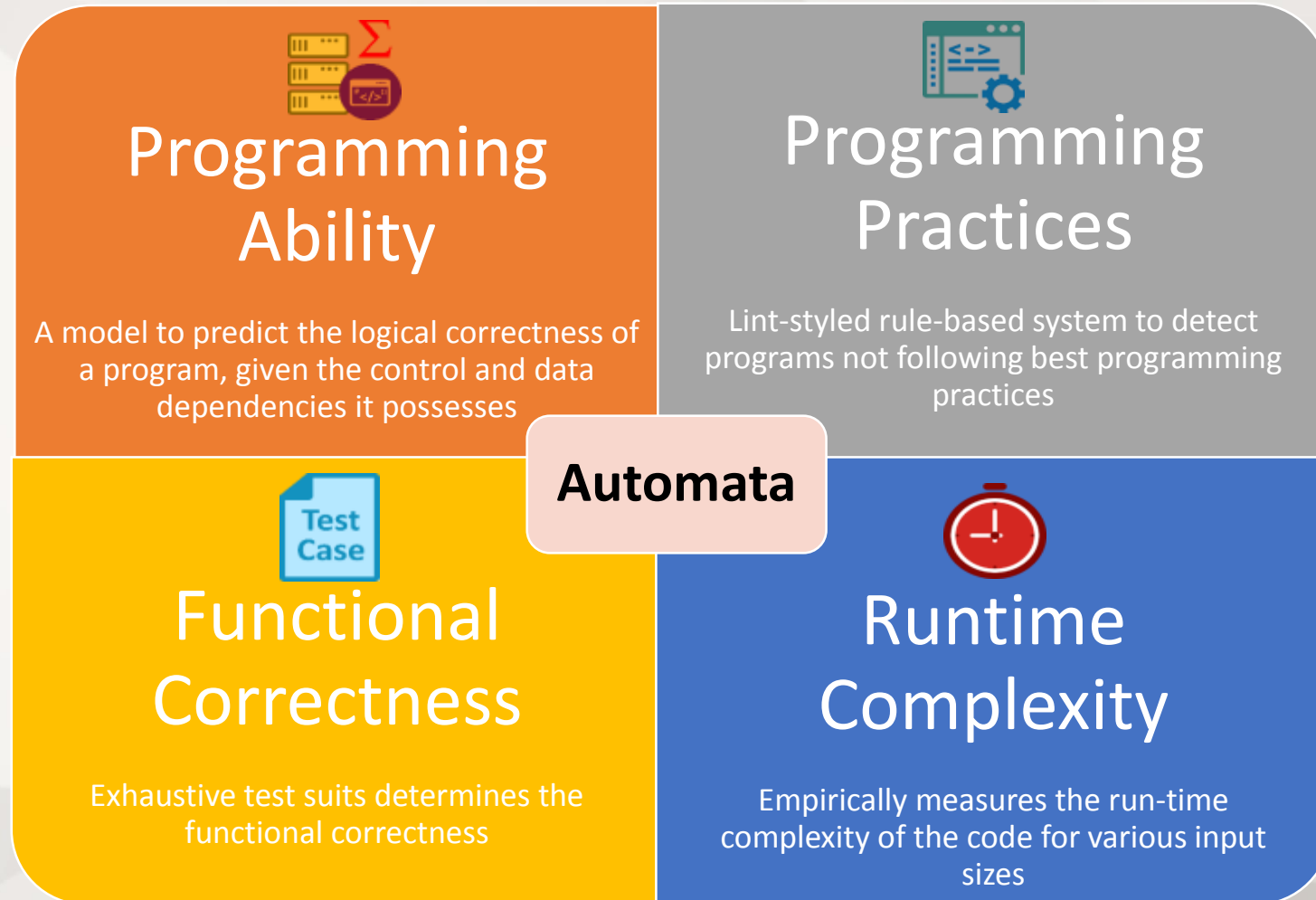
The tool can predict job suitability of a candidate by evaluating his/her coding ability on the following parameters:

- ❑ **Programming Ability** – evaluates the implementation of right algorithm using the control-structures, data dependencies and relevance of solution.
- ❑ **Programming Practices** – evaluated the code for the best practices implemented across industry while developing code which enhances its readability and usability.
- ❑ **Runtime complexity** – evaluated the runtime complexity by comparing to the runtime complexity of an optimized code.
- ❑ **Functional correctness** – measures the degree of code correctness based on the test-suite coverage by the developed code. The type of test cases evaluated are: Basic, Advanced and Edge.
- ❑ **Final score** – calculates overall score by weighing the candidate's performance in each of the 4 metrics.

Based on these parameters, the Automata Feedback Report predicts the skills required to develop algorithm in various coding jobs in the software/IT industry.

Programming Skills

Automated Programming Ability Evaluation



Programming Skills

Automated Programming Ability Evaluation

Automata – Aspiring Minds' Programming Ability Assessment:

- ❑ **Research driven holistic** evaluation that can be combined with other assessments.
- ❑ **Automated simulation** based test with easy option to compile, execute and save programs.
- ❑ Scoring on the basis of **standard rubrics** with very high correlation to different parameters.
- ❑ **Accurate prediction and scoring** of Programming Ability , Programming Practices, Runtime Complexity and Functional Correctness parameters.
- ❑ **Abridged problem statements** in candidate's choice of programming language – C, C++ and java.
- ❑ **Standardized and validated** approach leading to effective and efficient hiring.

National Status of Programming Skills

Rubric	Definition	Grand Total*
A4	Candidate is able to write functionally and logically correct code	2.21%
A3	Candidate is able to write functionally correct code with few anomalies	2.56%
A2	Candidate is not able to write functionally correct code	31.01%
A1	Candidate is not able to write compilable code	64.22%

* Distribution of test takers across various defined levels.

✓ Programming Ability rubrics are divided into four broad divisions on the basis of 4 sub-sections of Automata

National Status of Programming Skills

Rubric	Definition	Grand Total*
A4.4	Candidate is able to write functionally-logically correct, efficient & maintainable code	63.84%
A4.3	Candidate is able to write functionally-logically correct, efficient code. However, the code written is unmaintainable	8.98%
A4.2	Candidate is able to write functionally-logically correct, maintainable code. However, the code written is low on efficiency	25.22%
A4.1	Candidate is able to write functionally-logically correct code. However, the code written is low on efficiency & maintainability	1.97%

* Distribution of test takers across various defined levels.

✓ Functionally correct codes of level A4 are further categorized on the basis of attributes of written code

Observations

- ❑ Automata presents 2 problem statements closely related to on-job tasks and uses **automatic program evaluation engine** to grade the written code.
- ❑ We find only **2.21%** engineers lie at the **level of A4**. These candidates show the ability to develop a logically correct algorithm for the given problem. These candidates are able to clear all the test cases and implement average-high time-space complexity and indentation styles. This is acceptable because these candidates have no or minimum industrial exposure. With **structured guidance, these candidates can develop fully functional codes** as per the corporate requirements.
- ❑ Only **2.56% engineers** show a level of A3. At this level, the code written has a few inadvertent errors which causes some test cases to fail. Such errors may occur due to incorrect implementation of advance functionality and partial data dependencies There is a clear need for a much **better training curriculum for programming ability concepts** to understand how to develop optimized code.
- ❑ The coding capability of engineers need substantial improvement. **Given the high growth rate in jobs in software sector**, there is an urgent need of interventions to improve the programming skillset of engineers.

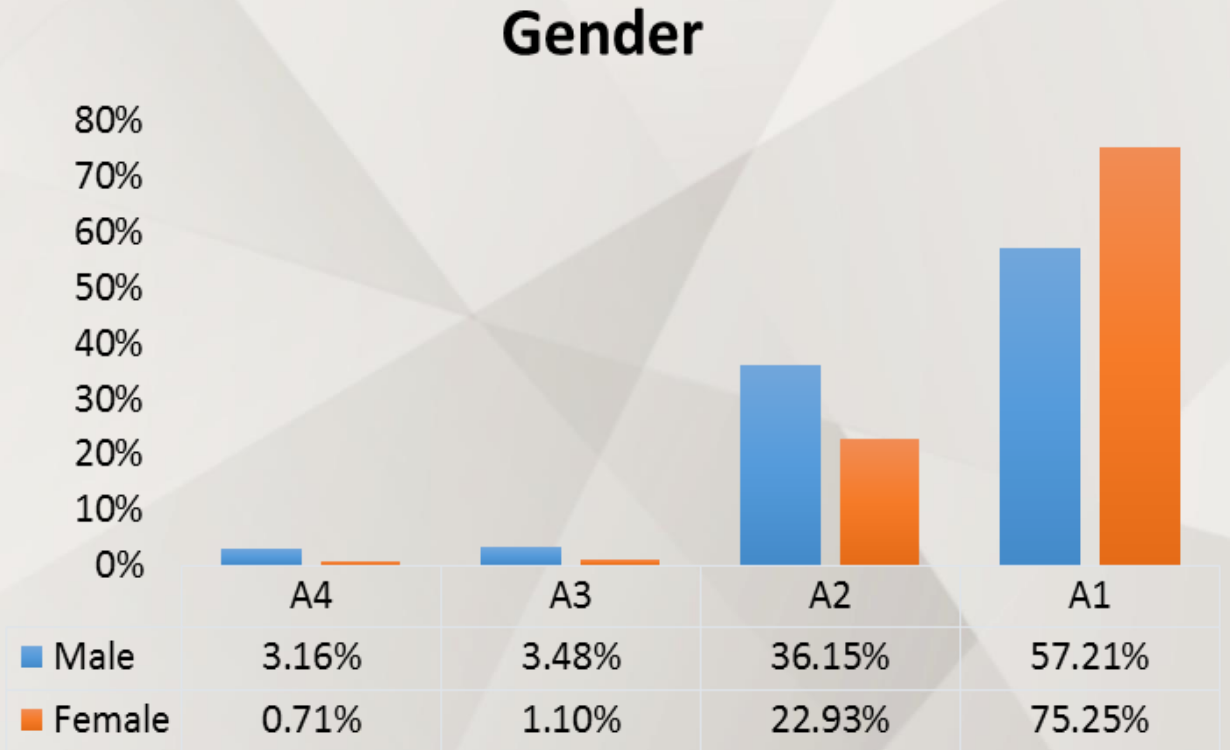
Observations

- ❑ IT software companies care for how well candidates can understand the given problem and develop its algorithm **covering all use-cases possible** with best programming practices for the ease of scalability and debugging.
- ❑ This indicates that even though candidate are able to develop correct logic for the problem, they struggle to write a maintainable, efficient code with proper implementation of logic in a **time bound scenario**.
- ❑ We observe that the **maximum gap is in the Programming Ability and Programming Practices** of the candidates. **Only 9.21% and 15.63% candidates can apply machine learning and maintainability & usability principles**, respectively, that renders an ideal code.
- ❑ This warrants a **greater emphasis to programming skills** at the university level to impart better knowledge and broader thinking required to deploy real-time efficient codes.

Programming Skill

Gender Comparison

Programming Skill	Rubric	Gender	
		Male	Female
A4	Functionally and logically correct code	3.16%	0.71%
A3	Functionally correct code with few anomalies	3.48%	1.10%
A2	Functionally incorrect code	36.15%	22.93%
A1	Syntax Error	57.21%	75.25%



- ✓ The analysis unveils that males have better programming skills than females.
- ✓ Only 25% of females are able to write a compilable code and less than 1% can write logically & functionally correct code.

Mean Scores

Gender comparison

Gender	Mean score*			
	Programming Ability score	Programming Practices score	Runtime Complexity score**	Final score
Male	17.75	18.66	20.57	15.14
Female	8.20	9.00	13.00	6.70
Difference***	9.55	9.66	7.57	8.44

*All scores in Automata modules are on a scale of 0 to 100

**Complexity score distribution shown for those test takers who wrote a compilable code

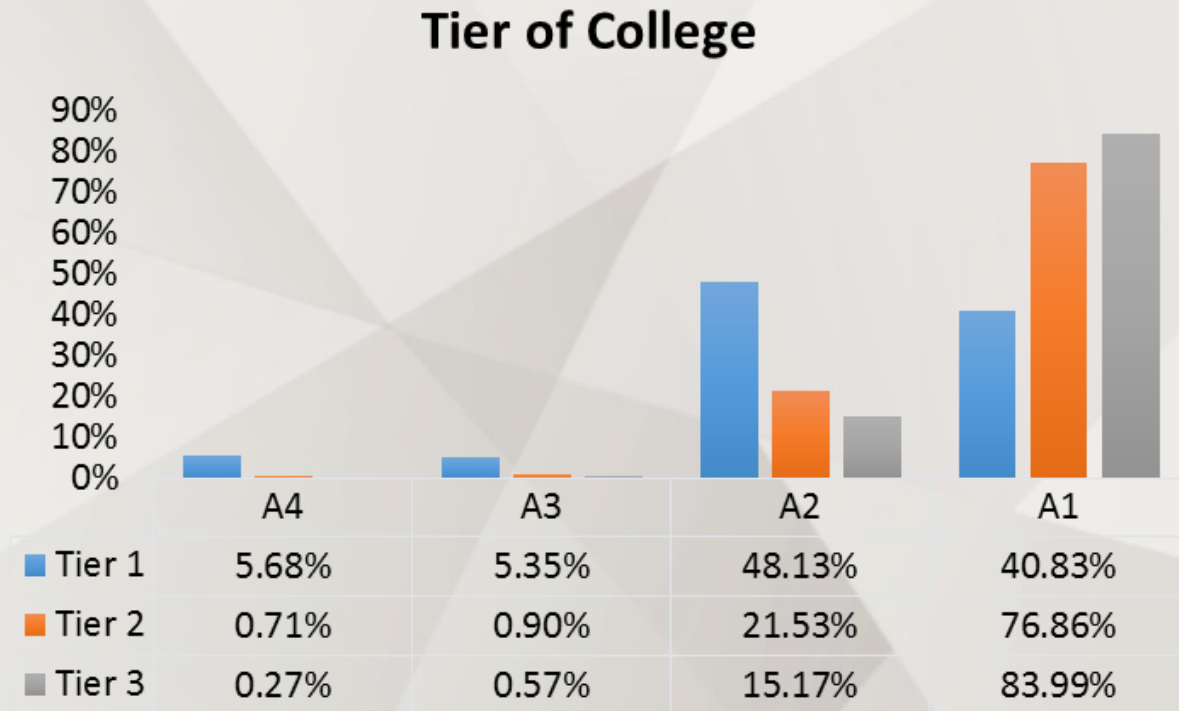
***Difference = Male scores- Female scores

- ✓ Similar score difference is observed between males and females across all subsections.
- ✓ Maximum gap is observed in Programming Practices followed by Programming Ability subsection.
- ✓ In summary, females lack the skillset of logically correct code with reusable elements.

Programming Skill

Comparison by College Tier

Programming Skill	Rubric	College Tier		
		Tier 1	Tier 2	Tier 3
A4	Functionally and logically correct code	5.68%	0.71%	0.27%
A3	Functionally correct code with few anomalies	5.35%	0.90%	0.57%
A2	Functionally incorrect code	48.13%	21.53%	15.17%
A1	Syntax Error	40.83%	76.86%	83.99%



- ✓ The quality of programming skills is influenced by the tier of the college. As expected the coding ability of candidates becomes worse in lower tier colleges.
- ✓ More degradation is observed from tier 1 to tier 2 colleges as compared to that from tier 2 to tier 3 colleges.

Mean Scores

Comparison by College Tier

College Tier	Mean score*			
	Programming Ability score	Programming Practices score	Runtime Complexity score**	Final score
1	26.17	25.58	23.36	22.81
2	7.73	8.63	13.28	6.19
3	5.07	5.57	12.76	4.02
Diff Tier 1 to Tier 2	18.44	16.95	10.08	16.62
Diff Tier 2 to Tier 3	2.66	3.06	0.53	2.17

*All scores in Automata modules are on a scale of 0 to 100

**Complexity score distribution shown for those test takers who wrote a compilable code

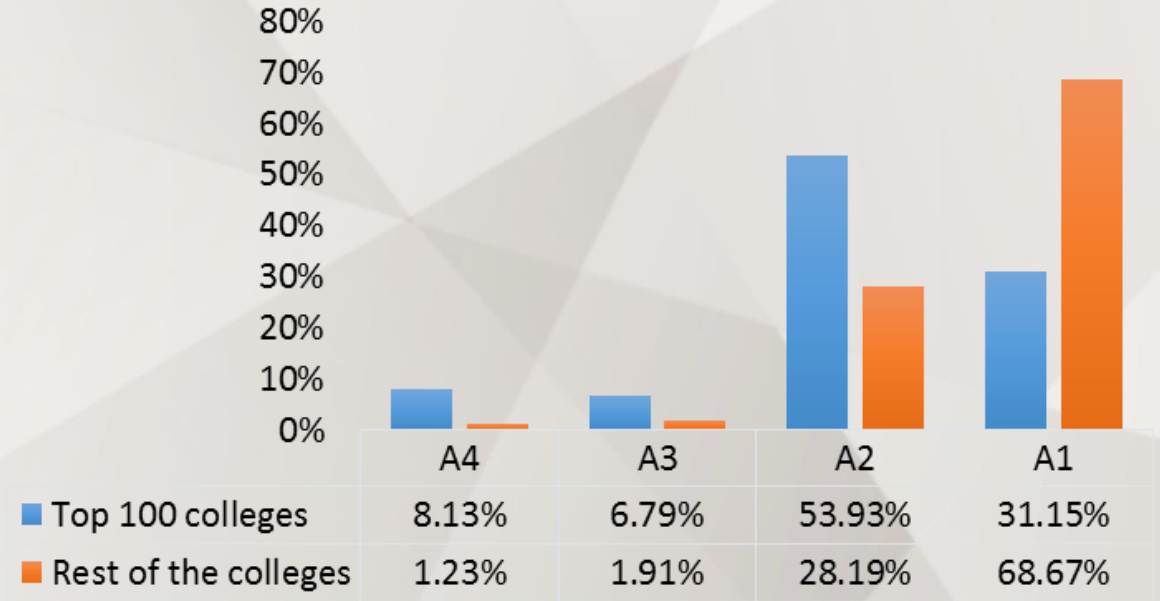
- ✓ Prominent score gap is observed in tier 1 and tier 2 colleges. This suggests training intervention is needed to improve programming skills in all aspects: logic, maintainability and efficiency.
- ✓ We find tier 2 and tier 3 colleges have comparable mean scores across all subsection where maximum gap is observed in subsection Programming Practices(code maintainability & reusability skills).
- ✓ In summary, candidates from tier 1 colleges dominate in programming skillset. And a tier 3 college candidate can be considered at par to a tier 2 college candidate.

Programming Skills

Comparison by Top 100 colleges vs Rest

Programming Skill	Rubric	College variance	
		Top 100 colleges	Rest of the colleges
A4	Functionally and logically correct code	8.13%	1.23%
A3	Functionally correct code with few anomalies	6.79%	1.91%
A2	Functionally incorrect code	53.93%	28.19%
A1	Syntax Error	31.15%	68.67%

Top 100 colleges vs rest



- ✓ The study shows that candidates from top 100 colleges have high proficiency of programming skills than those from remaining colleges.
- ✓ 31% candidates from top 100 colleges are not able to write a compilable code whereas from rest of the colleges only 31% are able to write compilable code.

Mean Scores

Comparison by Top 100 colleges vs Rest

Colleges	Mean score*			
	Programming Ability score	Programming Practices score	Runtime Complexity score**	Final score
Top 100 colleges	32.39	31.43	26.39	28.59
Rest of the colleges	11.31	12.55	15.34	9.31
Difference***	21.08	18.87	11.05	19.27

*All scores in Automata modules are on a scale of 0 to 100

**Complexity score distribution shown for those test takers who wrote a compilable code

***Difference = Top100 campuses – rest of the campuses

- ✓ The score trend matches the general perception that candidates from top 100 colleges generally possess superior programming skills than any other college.
- ✓ As observed previously, Programming Ability is the evident differentiating subsection.

Thank You!