**Content for Literature Review:**

**Courses:**

Complete 3-4 weeks in every course to formulate patterns and understand the following details:

* What are the key components and topics covered in existing programming courses
* How do universities structure their syllabi for such courses?
* Analysis of the typical learning outcomes and objectives of the competitive programming courses
* Analysis of the teaching methods and resources commonly used in these courses
* The effectiveness of the competitive programming
* Existing frameworks or metrics for measuring effectiveness of such courses
* The working and the efficiency of the platforms used for the course – languages and environments used, online platforms, coding competitions or challenges integrated into these courses
* The connection of the course with the ICPC
* Assessment methods employed to evaluate student performance in courses
* Strategies and activities used to engage and motivate students in competitive programming courses
* The review of the students who took the course

**Structuring:**

Heading, Title, Authors, Abstract, Introduction, Conclusion, Citation

Teaching method, difficulty level, languages, assignments. Topics, programming assignments

**Links and Titles of the Previous Research:**

**Motivate Competitive Programming:**

**Research Article 1:**

**Title:** Competitive programming and visualization technology as strategy to engage students in the modern educational process

**Link:** <https://library.iated.org/view/TRAYKOV2021COM>

|  |
| --- |
| Only Abstract is given; do not see any option to download |

**Research Article 2:**

**Title:** Competitive programming: A teaching methodology analysis applied to first-year programming classes

**Link:** [**http://library.lol/scimag/10.1109/fie43999.2019.9028518**](http://library.lol/scimag/10.1109/fie43999.2019.9028518)

|  |
| --- |
| * Link redirects to another paper but title also suggests paper is similar to Research Article 7 below |

**Research Article 3:**

**Title:** Competitive programming and gamification as strategy to engage students in computer science courses

**Link:** <https://www.revistaespacios.com/a18v39n35/18393511.html>

|  |
| --- |
| * Like many other papers here this paper discusses implementation of a combination of a competitive and gamified approach to computer science course; not directly relevant to out topic of research |

**Research Article 4:**

**Title:** Toward improving collaborative behavior during competitive programming assignments

**Link:** <https://sci-hub.ru/10.1109/EduHPC49559.2019.00014>

|  |
| --- |
| * Approach to integrate collaborative gamification techniques with the competitive elements of programming contests to create a balance. * Description of a forum design where students are graded on their activity: questions/answers/tips posted; mainly to encourage an environment of collaboration; presents pros and cons of having such a forum esp when students are already using other tools such as telegram etc. |

**Research Article 5:**

**Title:** Competitive programming in computational thinking and problem-solving education

**Link:** <https://onlinelibrary.wiley.com/doi/10.1002/cae.22610?af=R>

|  |
| --- |
| Will require access |

**Designing a Competitive Programming course for Students:**

**Research Article 1:**

**Title:** Competitive Programming in National University of Singapore

**Authors:** Steve and Felix Halim

**Link:** [https://citeseerx.ist.psu.edu/document repid=rep1&type=pdf&doi=3b5e09c5da225e120e4a3b22a6df6c14ef474a08](https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=3b5e09c5da225e120e4a3b22a6df6c14ef474a08)

|  |
| --- |
| * Year 3 (those with exp in NOI or IOI can be allowed from Year 1 with permission) * 13 teaching weeks \* 3-4 hours per week * Focus on efficient implementation of algorithms; done by programming contest examples * Syllabus of course is given * Use syllabus to gauge student’s/team’s strengths (scoring) * Lectures + local contests (4-5 with 4-6 problems each) * A sample test is given and a visualization of contest dynamics is shown * Advantages of a dedicated module:   >> distinguish the talented programmers  >> train them more intensively  >> give each other comparably strong ‘sparring partners’ to keep them motivated  >> top ones will perform better in ICPCs  >> A dedicated module also allows University administration to specifically allocate resources   * Challenges:   >> manpower: need for experienced instructor and TAs  >> students: getting top/grade-conscious students to join course; student’s initial strength uneven  >> assessment: producing original problem sets for local contests to avoid incorrect scores  >> course materials challenges: need for a course specific text book   * Addressing challenges   >> NUS has multiple ex-ICPC World Finalists as PhD students/Research Assistants + devoted professors/lectures sharing domain-specific knowledge  >> Advertises module University-wide at start of academic year to attract top programmers from all majors (apart from CS); ex-students advertise to juniors; coupled with benefits of representing in ICPC; two-tiered approach in class.  >> 1) original efforts by teaching staff 2) paraphrasing old problems 3) homework: students set problems for juniors  >> use recent ICPC and IOI problems on top of existing textbooks   * ICT Tools Used:   >> ACM ICPC Live Archive – Lib of problems; sample problems/homeworks  >> UVa Online Judge  >> Hunting UVa Problems – filters problems from UVa; generate ranking; self-learning  >> Methods to Solve – hints to solve approx. 40 percent problems in UVa  >> UVaToolkit – finding problems of similar types  >> π Algorithmist similar to Methods to Solve and UVaToolkit; usage of wiki  >> Mooshak – arranging local contests; data hosted locally; Universidade do Porto  >> TopCoder – recent programming contest techniques  >> Igor’s Code Archive – a compilation of codes   * Details abt & features of Uva tools created and ACM ICPC Live Archive; might be helpful to explore in depth later * Interested readers who want to set up similar module in their University can contact the authors directly to obtain some PowerPoint slides, sample codes, and sample contests data prepared by the authors. |

**Research Article 2:**

**Title:** What is the Competitive Programming Curriculum

**Link:** <https://helda.helsinki.fi/bitstream/handle/10138/352514/v16_2022_35_42.pdf?sequence=1>

|  |
| --- |
| * In this paper we go through the “Algorithms and Complexity” knowledge area in the ACM/IEEE curriculum guidelines and determine which of the topics can be learned through competitive programming. After that, we discuss in detail some topics that are different in competitive programming and university courses. * Not very relevant to our research; discussion mainly on comparison of topics covered in competitive programming vs. university courses |

**Research Article 3:**

**Title:** A Competitive Programming Approach to a University Introductory Algorithms Course

**Link:** <https://ioinformatics.org/journal/v11_2017_87_92.pdf>

|  |
| --- |
| * Not very relevant to our research; describes a data structures and algorithms course that was initially primarily theoretic and a more practical approach was adopted testing students on their programming skills; sounds much like HackerRank |

**Research Article 4:**

**Title:** A Framework for Personalized Competitive Programming Training

**Link:** <https://ieeexplore.ieee.org/document/8424620> , <https://sci-hub.ru/10.1109/ITHET.2018.8424620>

|  |
| --- |
| * The paper presents a dynamic framework to assist students in their preparation for the ICPC; Not very relevant to our current research |

**Research Article 5:**

**Title:** Learning Analytics in Competitive Programming Training Systems

**Link:** <https://sci-hub.ru/10.1109/iV.2018.00061>

|  |
| --- |
| * Not relevant; analysis of data collected from two platforms, OII-training and DIGIT, one for students and other for teachers; maily to emphasize similarity between the two groups when dealing with programming contest training. |

**Research Article 6:**

**Title:** Competitive programming: A teaching methodology analysis applied to first-year programming classes

**Link:** <http://library.lol/scimag/10.1109/fie43999.2019.9028518> (downloadable pdf)

|  |
| --- |
| * Link redirects to another paper but title also suggests paper is similar to Research Article 7 below |

**Research Article 7:**

**Title:** Teaching data structures using competitive games

**Link:** <https://sci-hub.ru/10.1109/te.2004.825053>

|  |
| --- |
| * Didn’t read full paper; Abstract indicates main focus is on using gamification in data structure courses |

**Research Article 8:**

**Title:** Early Introduction of Competitive Programming

**Link:** <https://www.mclibre.org/descargar/docs/revistas/olympiads-in-informatics/olympiads-in-informatics-02-en-200808.pdf#page=149>

|  |
| --- |
| The paper mainly talks abt implementing Mooshak platform in programming courses. A more thorough read needed |

Note: Research Article 4 along with other papers in this category suggests a gamified approach when preparing students for competitive programming, with reward systems like badges and a system that selects appropriate problems for the students depending on their capabilities and learning process

**Measuring the Efficiency and Effectiveness of the Course:**

**Research Article 1:**

**Title:** Good Practice Principles in an Undergraduate Blended Course Design

**Link:** <https://www.proquest.com/openview/39511ce8055f87bb39cc8fc122ab665d/1?pq-origsite=gscholar&cbl=1056401>

|  |
| --- |
| Only preview viewable  Abstract suggests a mode of instruction that is a mix of inperson and online, involving face to face instruction + online supplements to assist |

**Research Article 2:**

**Title:** Principles for Good Practice in Undergraduate Education: Effective Online

Course Design to Assist Students’ Success

**Link:** <https://uscdmc.sc.edu/about/offices_and_divisions/cte/instructional_design/docs/principles_good_practice_undergraduate_education_crews.pdf>

|  |
| --- |
| Title suggests online course design; not relevant to our research |

**Research Article 3:**

**Title:** Designing for learning: What are the essential features of an effective online course?

**Link:** <https://ajet.org.au/index.php/AJET/article/view/1926>

|  |
| --- |
| Title suggests online course design; not relevant to our research |

**Research Article 4:**

**Title:** Competitive programming: A teaching methodology analysis applied to first-year programming classes

**Link:** <https://ieeexplore.ieee.org/document/9028518>

|  |
| --- |
| Title suggests irrelevant; discussed is competitive programming as a teaching methodology not a specific course designed to prepare students for likes of ICPC |

**Research Article 5:**

**Title:** Novel Teaching Strategies for Lab-Centric Courses: Case Study of Programming Course

**Link:** <https://sci-hub.ru/10.1109/T4E.2018.00038>

|  |
| --- |
| The paper discusses some teaching approaches for lab-centric approaches: POE-I, BET, BEST, BESDIT. The main idea is to have theory and the laboratory sessions blended together. The strategies can be helpful to explore for implementation in the new course. |

**Platforms for Competitive Programming:**

**Research Article 1:**

**Title:** A study of competitive programming platform with its need and benefits

**Link:** <https://www.indianjournals.com/ijor.aspx?target=ijor:jims8i&volume=10&issue=2&article=001>

|  |
| --- |
| Abstract suggests quite relevant. Need access to open paper |

**Research Article 2:**

**Title:** Facilitating Course Assessment with a Competitive Programming Platform

**Link:** <https://dl.acm.org/doi/abs/10.1145/3287324.3287511>

|  |
| --- |
| Abstract suggests the paper discusses the impacts (pros and cons) of using a competitive programming platform for an algorithms course. Since our course is on competitive programming itself this paper is not relevant to our research |

**Research Article 3:**

**Title:** A gamified web-based system for computer programming learning

**Link:** <https://www.sciencedirect.com/science/article/pii/S2666920X21000230>

|  |
| --- |
| The paper describes a system, 2TSW that is designed to implement gamification strategies to computer programming courses. Not very relevant to our research. |

**Research Article 4:**

**Title:** OJPOT: online judge & practice oriented teaching idea in programming courses

**Link:** [https://www.tandfonline.com/doi/abs/10.1080/03043797.2015.1056105?\_gl=1\*ju7ssx\*\_ga\*MTg4MDMwMDQ1My4xNjg2MTcyODMw\*\_ga\_0HYE8YG0M6\*MTY4NjE3MjgzMS4xLjEuMTY4NjE3MjkyNy4wLjAuMA..&\_ga=2.39934689.2117184370.1686172830-1880300453.1686172830](https://www.tandfonline.com/doi/abs/10.1080/03043797.2015.1056105?_gl=1*ju7ssx*_ga*MTg4MDMwMDQ1My4xNjg2MTcyODMw*_ga_0HYE8YG0M6*MTY4NjE3MjgzMS4xLjEuMTY4NjE3MjkyNy4wLjAuMA..&_ga=2.39934689.2117184370.1686172830-1880300453.1686172830)

|  |
| --- |
| Require access |

**Book 1:**

**Title:** Guide to Competitive Programming

**Link:** <https://link.springer.com/book/10.1007/978-3-030-39357-1>

ACM/IEEE (2013). Curriculum Guidelines for Undergraduate Programs in Computer Science. Available online at: <https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf> Cormen