CS Standards Crosswalk: Summary Checklist
CSTA K-12 Computer Science Standards and Oracle Java Programming (2014)

CSTA Website	http://csta.acm.org/Curriculum/sub/K12Standards.html
Oracle Website	https://academy.oracle.com/oa-web-introcs-curriculum.html
Oracle Contact	academy_ww@oracle.com

Level 3 (recommended for grades 9–12) Applying concepts and creating real-world solutions

Level 3 is divided into three discrete courses, each of which focuses on different facets of computer science as a discipline. Throughout these courses, students can master more advanced computer science concepts and apply those concepts to develop virtual and real-world artifacts.

Level 3A: (recommended for grades 9 or 10) Computer Science in the Modern World

This course is recommended for all students. Its goal is to solidify students' understanding of computer science principles and practices so that they can make informed choices and use appropriate computational tools and techniques in whatever career they decide to pursue. They should also appreciate the breadth of computing and its influence in almost every aspect of modern life.

✓ = CSTA standard is met by one or a combination of Java Programming objectives.

C = one or more Java Programming objectives contribute to but are not sufficient for meeting the CSTA standard.

CSTA Standard: *Level 3A (Grade 9 or 10)*	Course Meets Standard
Strand: Computational Thinking	
Use predefined functions and parameters, classes and methods to divide a complex problem into simpler parts.	
Describe a software development process used to solve software problems (e.g., design, coding, testing, verification).	
Explain how sequence, selection, iteration, and recursion are building blocks of algorithms.	
Compare techniques for analyzing massive data collections.	
Describe the relationship between binary and hexadecimal representations.	
Analyze the representation and trade-offs among various forms of digital information.	
Describe how various types of data are stored in a computer system.	
Use modeling and simulation to represent and understand natural phenomena.	✓
Discuss the value of abstraction to manage problem complexity.	
Describe the concept of parallel processing as a strategy to solve large problems.	
Describe how computation shares features with art and music by translating human intention into an artifact.	
Strand: Collaboration	
Work in a team to design and develop a software artifact.	
Use collaborative tools to communicate with project team members (e.g., discussion threads, wikis, blogs, version control, etc.).	
Describe how computing enhances traditional forms and enables new forms of experience, expression, communication, and collaboration.	
Identify how collaboration influences the design and development of software products.	
Strand: Computing Practice and Programming	
Create and organize web pages through the use of a variety of web programming design tools.	
Use mobile devices/ emulators to design, develop, and implement mobile computing applications.	

CSTA Standard: *Level 3A (Grade 9 or 10)*	Course Meets Standard
Use various debugging and testing methods to ensure program correctness (e.g., test cases, unit testing, white box, black box, integration testing)	✓
Apply analysis, design, and implementation techniques to solve problems (e.g., use one or more software lifecycle models).	✓
Use Application Program Interfaces (APIs) and libraries to facilitate programming solutions.	✓
Select appropriate file formats for various types and uses of data.	
Describe a variety of programming languages available to solve problems and develop systems.	
Explain the program execution process.	**C**
Explain the principles of security by examining encryption, cryptography, and authentication techniques.	
Explore a variety of careers to which computing is central.	
Describe techniques for locating and collecting small and large-scale data sets.	
Describe how mathematical and statistical functions, sets, and logic are used in computation.	
Strand: Computers and Communication Devices	
Describe the unique features of computers embedded in mobile devices and vehicles (e.g., cell phones, automobiles, airplanes).	
Develop criteria for purchasing or upgrading computer system hardware.	
Describe the principal components of computer organization (e.g., input, output, processing, and storage).	
Compare various forms of input and output.	**C**
Explain the multiple levels of hardware and software that support program execution (e.g., compilers, interpreters, operating systems, networks).	
Apply strategies for identifying and solving routine hardware and software problems that occur in everyday life.	
Compare and contrast client-server and peer-to-peer network strategies.	
Explain the basic components of computer networks (e.g., servers, file protection, routing, spoolers and queues, shared resources, and fault-tolerance).	
Describe how the Internet facilitates global communication.	
Describe the major applications of artificial intelligence and robotics.	
Strand: Community, Global, and Ethical Impacts	
Compare appropriate and inappropriate social networking behaviors.	
Discuss the impact of computing technology on business and commerce (e.g., automated tracking of goods, automated financial transactions, e-commerce, cloud computing).	
Describe the role that adaptive technology can play in the lives of people with special needs.	
Compare the positive and negative impacts of technology on culture (e.g., social networking, delivery of news and other public media, and intercultural communication).	
Describe strategies for determining the reliability of information found on the Internet.	
Differentiate between information access and information distribution rights.	
Describe how different kinds of software licenses can be used to share and protect intellectual property.	
Discuss the social and economic implications associated with hacking and software piracy.	
Describe different ways in which software is created and shared and their benefits and drawbacks (commercial software, public domain software, open source development).	
Describe security and privacy issues that relate to computer networks.	
Explain the impact of the digital divide on access to critical information.	

Level 3B: (recommended for grades 10 or 11) Computer Science Concepts and Practices

This course is a more in-depth study of computer science and its relation to other disciplines, and contains a significant amount of algorithmic problem solving and related activities. One way to realize this course is by following the Computer Science Principles course (www.apcsprinciples.org). Students should complete this course with a clear understanding of the application of computational thinking to real-world problems.

✓ = CSTA standard is met by one or a combination of Java Programming objectives.

C = one or more Java Programming objectives contribute to but are not sufficient for meeting the CSTA standard.

CSTA Standard: *Level 3B (Grade 10 or 11)*	**Course Meets Standard**
Strand: Computational Thinking	
Classify problems as tractable, intractable, or computationally unsolvable.	
Explain the value of heuristic algorithms to approximate solutions for intractable problems.	
Critically examine classical algorithms and implement an original algorithm.	
Evaluate algorithms by their efficiency, correctness, and clarity.	
Use data analysis to enhance understanding of complex natural and human systems.	
Compare and contrast simple data structures and their uses (e.g., arrays and lists).	
Discuss the interpretation of binary sequences in a variety of forms (e.g., instructions, numbers, text, sound, image).	
Use models and simulations to help formulate, refine, and test scientific hypotheses.	
Analyze data and identify patterns through modeling and simulation.	
Decompose a problem by defining new functions and classes.	**C**
Demonstrate concurrency by separating processes into threads and dividing data into parallel streams.	
Strand: Collaboration	
Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.	
Demonstrate the software life cycle process by participating on a software project team.	
Evaluate programs written by others for readability and usability.	
Strand: Computing Practice and Programming	
Use advanced tools to create digital artifacts (e.g., web design, animation, video, multimedia).	
Use tools of abstraction to decompose a large-scale computational problem (e.g., procedural abstraction, object-oriented design, functional design).	**C**
Classify programming languages based on their level and application domain.	
Explore principles of system design in scaling, efficiency, and security.	
Deploy principles of security by implementing encryption and authentication strategies.	
Anticipate future careers and the technologies that will exist.	
Use data analysis to enhance understanding of complex natural and human systems.	
Deploy various data collection techniques for different types of problems.	
Strand: Computers and Communication Devices	
Discuss the impact of modifications on the functionality of application programs.	
Identify and describe hardware (e.g., physical layers, logic gates, chips, components).	
Identify and select the most appropriate file format based on trade-offs (e.g., accuracy, speed, ease of manipulation).	
Describe the issues that impact network functionality (e.g., latency, bandwidth, firewalls, server capability).	
Explain the notion of intelligent behavior through computer modeling and robotics.	
Strand: Community, Global, and Ethical Impacts	
Demonstrate ethical use of modern communication media and devices.	
Analyze the beneficial and harmful effects of computing innovations.	

CSTA Standard: ***Level 3B (Grade 10 or 11)***	**Course Meets Standard**
Summarize how financial markets, transactions, and predictions have been transformed by automation.	
Summarize how computation has revolutionized the way people build real and virtual organizations and infrastructures.	
Identify laws and regulations that impact the development and use of software.	
Analyze the impact of government regulation on privacy and security.	
Differentiate among open source, freeware, and proprietary software licenses and their applicability to different types of software.	
Relate issues of equity, access, and power to the distribution of computing resources in a global society.	

Level 3C (recommended for grades 11 or 12) Topics in Computer Science

This is an elective course that provides depth of study in one particular area of computing.

Course Leading to Industry Certification: Such a course is primarily geared toward students planning on entering the workforce, continuing their education in a post-secondary technical school, or entering a two-year college Associates of Applied Science program. Students taking this course should have completed Levels 1 and 2, and typically the Level 3A courses. Students who complete certification courses should be encouraged to take the corresponding exam as proof of acquired knowledge.

Oracle states that those who successfully complete its Java Programming course (as well as having mastered the prerequisite competencies in the Java Fundamentals course) will possess strong foundational knowledge for **Oracle Java SE 7 Programmer I certification.**

Therefore, CSTA has determined that the Java Programming course meets the definition of a Level 3C course.