

TEALS Program

[Home](#) | [Curriculum Map](#)

0.1 Aligning TEALS APCSA Teaching Resources with the BC Curriculum

This course focuses on object-oriented programming and problem solving in Java. The intended use is for schools offering AP College Board's Computer Science A course. For schools in BC, Canada, the course can be adapted to align with BC's grade 12 [Computer Programming](#) or [Computer Science](#) courses. Since 2018, our community partner BCIT has been supporting the curriculum alignment work. Here are highlights of the changes:

- Added Culture Days
- Created alternative, open-ended unit projects: [Unit 1](#), [Unit 3](#), [Unit 4](#) and [Unit 5](#)
- Incorporated BC curriculum's core competencies: communication, creative and critical thinking, personal and social responsibilities
- Included components of BC courses learning standards: big ideas, curricular competencies and content

0.2 For Computer Programming 12 (Applied Design, Skills and Technologies)

0.2.0.1 Curricular Competencies

- Curricular competencies under Applied Skills, such as ideating, prototyping and testing, are incorporated into the alternative projects and throughout the course.
- Activities involved conducting user-centred research are added to the course.

0.2.0.2 Content

- Students learn advance programming structure, documentation, error handling and debugging, etc.
- Throughout the course, students are encouraged to collaborate. Students are also given opportunities to develop interpersonal skills with Culture Day activities such as [Student Research Project and Presentation](#) and [Interviewing an IT Professional](#)
- In the open-ended projects, students apply design thinking skills and programming concepts.

0.3 For Computer Science 12 (Mathematics)

0.3.0.1 Curricular Competencies

- Reasoning and analyzing: Students explore, analyze, and apply mathematical ideas and computer science concepts in various exercises and projects in the course.
- Understanding and solving: Throughout the course, students represent computer science ideas in concrete, pictorial, and symbolic forms, such as writing pseudocode or creating diagrams.
- Communicating and representing: Students have opportunities, through various lesson activities, explain and justify computer science ideas. Teaching strategies such as Think-Pair-Share or pair-programming are encouraged in the teaching guides

- Connecting and reflecting: Students have opportunities throughout the course to reflect on mathematical and computational thinking

0.3.0.2 Content

- In Unit 3 [Calculator Project](#), students will apply mathematics in situational contexts.
- In Unit 4 [Open-Ended Programming Project](#), students have the option to develop their own encryption/decryption program.

0.4 Pacing

This course is originally designed for use in 50-min long classes. Below please see a sample pacing guide for using the course for 75-80 min long classes, with the assumption that there are about 85 instructional days for the course.

0.4.0.1 Unit 1: Programming & Java (8 days)

- Day 1: 1.01, 1.02
- Day 2: 1.03, 1.04
- Day 3: 1.05, 1.06
- Day 4-6: Project 1.07b, 1.08
- Day 7: Review
- Day 8: Unit 1 Test

0.4.0.2 Unit 2: Working with Data & Basic Control Flow (10 days)

- Day 1: 2.00, 2.01
- Day 2: 2.02, 2.03
- Day 3: 2.03, 2.04
- Day 4: 2.05
- Day 5: 2.06
- Day 6-7: 2.07, 2.08
- Day 8: 2.09, 2.10
- Day 9: Review
- Day 10: Unit 2 Test

0.4.0.3 Unit 3: Advanced Data & Control Flow (17 days)

- Day 1: 3.00, 3.01
- Day 2: 3.02, 3.03
- Day 3: 3.04
- Day 4: 3.05, 3.06
- Day 5: 3.07, 3.08
- Day 6: 3.09
- Day 7: 3.10
- Day 8: 3.11, 3.12
- Day 9: 3.13, 3.14
- Day 10: 3.16
- Day 11-15: Project 3.18b
- Day 16: Review
- Day 17: Unit 3 Test

0.4.0.4 Unit 4: Arrays, Lists, & Files (13 days)

- Day 1: 4.00, 4.01
- Day 2: 4.02
- Day 3: 4.03
- Day 4: 4.04, 4.05
- Day 5: 4.06
- Day 6: 4.07, 4.08
- Day 7-11: Project 4.09b
- Day 12: Review
- Day 13: Unit 4 Test

0.4.0.5 Unit 5: Object-Oriented Programming (11 days)

- Day 1: 5.00, 5.01
- Day 2: 5.02
- Day 3: 5.03
- Day 4: 5.04, 5.05
- Day 5-9: Project 5.06b
- Day 10: Review
- Day 11: Unit 5 Test

0.4.0.6 Unit 6: Inheritance & Polymorphism (12 days)

- Day 1: 6.00, 6.01
- Day 2: 6.01, 6.02
- Day 3: 6.03
- Day 4: 6.04
- Day 5: 6.05, 6.06
- Day 6-10: Project 6.07, 6.08
- Day 11: Review
- Day 12: Unit 6 Test

0.4.0.7 Unit 7 + 8: Algorithms: Recursion, Searching & Sorting (8 days)

- Day 1: 7.00, 7.01
- Day 2: 7.02
- Day 3: 8.01
- Day 4: 8.02
- Day 5: 8.03
- Day 6: 8.04, 8.05
- Day 7: Review
- Day 8: Unit 7 + 8 Test

0.4.0.8 Culture Days (2-3 days)