



MADLY PRODUCTIVE PATHWAYS

PROJECT-BASED CTE CURRICULUM
FOR BUSINESS & TECH CLASSROOMS



TERMS OF USE

Thank you for supporting independent curriculum design.

By purchasing this resource, you are agreeing that the contents are the property of Be Madly Productive and licensed to you only for classroom/ personal use as a single user. I retain the copyright and reserve all rights to this product. Please remember to [leave feedback](#) so you will earn TPT credits, which may be applied to your future purchases.

✓ DO:



Use free and purchased items for your own classroom students/personal use.



Reference this product in blog posts, at seminars, PD, workshops, or other such venues. ONLY if both credit is given to myself as the author, AND a link back to my store is included in the presentation.



Purchase licenses at a great discount for other teachers to use this resource.

✗ DON'T:



Claim this work as your own, alter the files in any way, or remove copyright / watermark.



Sell the files or combine them into another unit for sale / free.



Post this for sale / free elsewhere on the internet (including Google Doc links on blogs).



Make copies of purchased items to share with others (violation of the TOU and the law).

≡ STAY CONNECTED! ≡



Follow the Store!

[Follow the store](#) for new resources, sales, and discounts!



Join the Email List

[Sign up HERE](#) for exclusive freebies, teaching tips, and resource updates to your inbox!



Check Out Bundles!

Save time, money, and ENERGY with the full year bundles!



BUSINESS MINDS. TECH SKILLS. ENDLESS POSSIBILITIES.
Preparing students today. Leading tomorrow.





MADLY PRODUCTIVE PATHWAYS

YOU'RE ALL SET!

Your lesson files are in the folders beside this guide. Here is everything else worth knowing.

LIFETIME ACCESS

Your download link never expires.

Re-open the link from your confirmation email any time to download this resource again.

Updates are free, forever.

When this resource is revised for a new AP framework or a new school year, your copy updates with it at no charge.

NEED A HAND?

Email a real person.

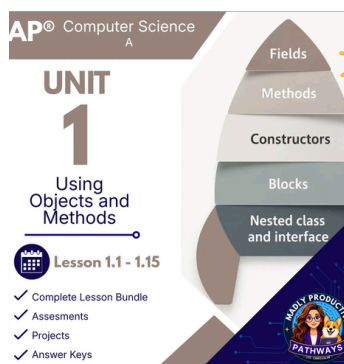
hello@madlyproductivepathways.com

Spotted a typo, or an answer key that looks off?

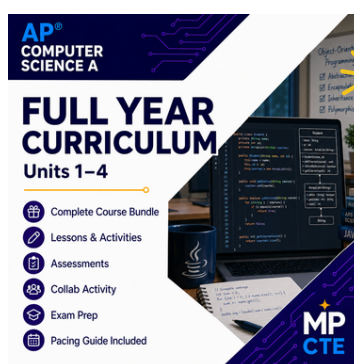
Tell us and we will fix it and re-issue the file. Feedback from the classroom is how this curriculum gets better.

⇒ BUILD YOUR CURRICULUM TOOLKIT ⇒

Check out these matching resources to save time and facilitate meaningful learning.



[Click to see the
Unit 1 Curriculum](#)



[Click to see the
Full Year Curriculum](#)



JOIN THE EMAIL LIST FOR
EXCLUSIVE FREEBIES AND PROJECT-
BASED TEACHING TIPS!



SCAN TO JOIN

Stay in the loop with
new resources, sales,
and classroom ideas



PROJECT-BASED. REAL-WORLD. **FUTURE-READY.**

Topic 1.1: Introduction to Algorithms, Programming, and Compilers

LO: 1.1.A, 1.1.B, 1.1.C | Skill: 1.A | Scenario: Algorithms in everyday life — then your first Java program

Welcome to Topic 1.1

This is the teacher's command-center document for Topic 1.1: Introduction to Algorithms, Programming, and Compilers — the first lesson of AP Computer Science A. Read this first before printing or assigning anything.

Topic	1.1 — Introduction to Algorithms, Programming, and Compilers
Unit	1: Using Objects and Methods
Estimated Time	2 class periods (CED recommendation, 45 min each)
Learning Objectives	1.1.A, 1.1.B, 1.1.C
Skill Focus	Skill 1.A
Lesson Theme	Algorithms in everyday life — then your first Java program
Original Scenario	Riverside Coding Club (original fictional scenario — no protected content)
Mini-FRQ Format	Algorithm + error analysis — 2 Java sources, Parts A & B (5 pts)
Exam Alignment	Equivalent to a first-semester college intro-programming course (Java)
Teacher Prep	Across 2 days — print Guided Notes, Worksheet, Scenario, Collaboration Activity, Exit Ticket, MCQs, Mini-FRQ. Slides for projection. No computers required for Day 1.

What students will know + do by the end

- Define an algorithm as a step-by-step process, and represent one in everyday life using written steps or a diagram.
- Explain that sequencing means the steps run one at a time, in order.
- Describe how source code becomes a running program: written in an editor or IDE, then checked by a compiler before it runs.
- Explain that the compiler catches some errors that must be fixed before the program will run.
- Identify and tell apart the three kinds of programming errors — syntax, logic, and run-time (including exceptions) — and how each is discovered.

Deliverables (this topic ships 24 files)

Every file is print-ready and audit-verified. Three handouts also ship as editable Word (.docx) twins. Folder paths shown.

File	Where it lives
READ FIRST	00 Start Here / 00_READ_FIRST_Teacher_Guide.pdf
Lesson Plan (2-day)	00 Start Here / 1.1_Lesson_Plan.pdf
Standards Map	00 Start Here / 1.1_Standards_Map.pdf
Slide Deck	01 Present / 1.1_Slides.pptx
Bell Ringer (Day 1)	01 Present / 1.1_Bell_Ringer.pdf
Guided Notes (Student) .pdf + .docx	02 Student Handouts / 1.1_Guided_Notes_Student.*
Vocab Slip	02 Student Handouts / 1.1_Vocab_Slip.pdf
Application Worksheet .pdf + .docx	02 Student Handouts / 1.1_Worksheet_Student.*
Hands-On Lab (Type & Run) .pdf + .docx	02 Student Handouts / 1.1_Lab_Type_and_Run_Student.*
Scenario (Student)	03 Scenario / 1.1_Scenario_Riverside_Coding_Club_Student.pdf
Collaboration Activity .pdf + .docx	04 Collaboration Activity / 1.1_Collab_Algorithms_and_Bugs.*
Exit Ticket	05 Assessment / 1.1_Exit_Ticket.pdf
MCQ Practice Set	05 Assessment / 1.1_MCQ_Practice_Set.pdf
Mini-FRQ	05 Assessment / 1.1_Mini_FRQ.pdf
Guided Notes Key	06 Answer Keys / 1.1_Guided_Notes_Key.pdf
Worksheet Key	06 Answer Keys / 1.1_Worksheet_Key.pdf
Scenario Teacher Key	06 Answer Keys / 1.1_Scenario_Riverside_Coding_Club_Teacher_Key.pdf
Exit Ticket Key	06 Answer Keys / 1.1_Exit_Ticket_Key.pdf
MCQ Key	06 Answer Keys / 1.1_MCQ_Key.pdf
Mini-FRQ Teacher Key	06 Answer Keys / 1.1_Mini_FRQ_Teacher_Key.pdf

Teacher prep checklist

- ☐ Print Guided Notes, Worksheet, Type-and-Run Lab, Scenario, Collaboration Activity, Exit Ticket, MCQs, Mini-FRQ — one per student.
- ☐ Project Slides (1.1_Slides.pptx). No projector audio needed (no embedded media).
- ☐ Day 1 needs no computers (concept + algorithm work). The Day-2 Type-and-Run lab needs a Java environment per student — a desktop IDE (BlueJ, IntelliJ, Eclipse), VS Code with the Java extension, or a free browser-based Java compiler. Set one up beforehand; if you have no devices, the lab can be skipped without breaking the lesson.
- ☐ Decide whether to assign the Mini-FRQ in class (Day 2 closer) or as homework.
- ☐ Pre-read the Scenario Teacher Key so you can guide the Day 2 debrief.

Where this topic fits

Within Unit 1 (Using Objects and Methods): Topic 1.1 is the on-ramp. Before students declare a single variable, they need the mental model of *algorithm* → *code* → *compile* → *run* and a vocabulary for what goes wrong. Topics 1.2 onward build the Java they will write inside that model.

Within the full course: the syntax / logic / run-time error distinction set up here is used in every later unit and is directly assessed on the free-response section, where students must explain why code will not compile or will not behave as intended.

TEACHING NOTE

Resist jumping straight to Java syntax. The single most valuable habit you can build today is the loop: write the algorithm in plain steps first, then translate to code, then expect to debug. Students who skip the algorithm step are the ones who later stare at a blank screen.