

ALGORITHMIC THINKING SUMMER EDUCATIONAL TOUR

JULY 26 - AUGUST 13, 2027

Dive into the world of computer programming with Python and utilize algorithmic thinking to build, solve and shape the future.

BUILD ALGORITHM THINKING: OPEN THE DOOR TO COMPUTER SCIENCE AND AI.

Algorithmic Thinking is a structured, beginner-friendly summer program for international high school students that introduces the foundations of computer science through algorithmic thinking. Students learn how computers 'think': how to break a problem into steps, test solutions, fix errors and improve efficiency.

Python is used as the practical doorway because it is widely used in modern fields such as machine learning, artificial intelligence, data science and software engineering. However, the program is not about memorizing syntax, it is about building durable skills: logic, discipline and computational problem solving.

Students learn through hands-on mini-projects, from math challenges and quizzes to beginner game mechanics with simple graphics and sound. The program ends with a final project showcase, giving students a portfolio-ready project and a clearer view of future pathways in CS, AI and the game industry.



PROGRAM OVERVIEW

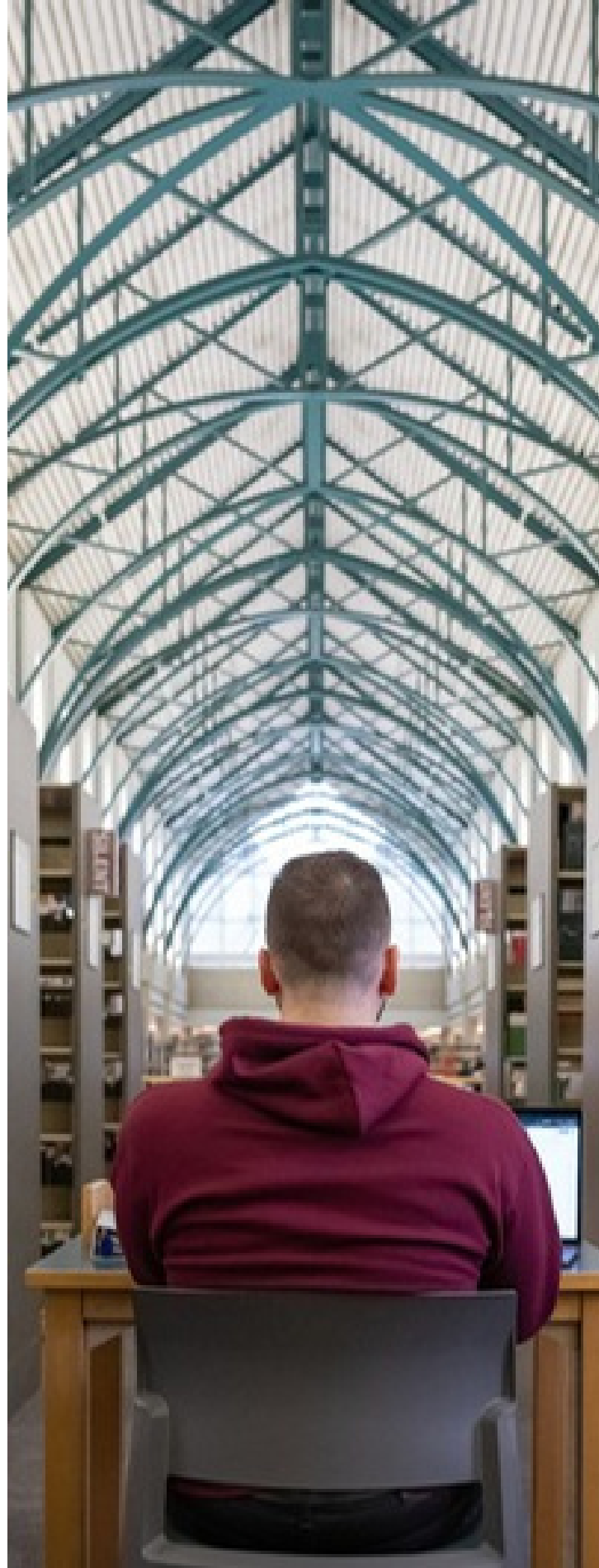
- A summer school program for young learners **ages 12+**
- One to three weeks of project-based and game-based learning by **prototyping an interactive game from scratch**.
- Classes hosted by **industry professionals** at a Canadian higher education institution located in Edmonton, Alberta.
- Hands-on use with **Python**, a real-world application as the doorway to future CS/AI/ML learning.
- **Certificate of Attendance** upon completion and final project showcase for building portfolio value.
- **No experience needed:** If students can use a computer and are curious, they are welcome and ready to attend.

**Study tours can last anywhere from 1-3 weeks, but are not required to last the full 3 weeks.*

WHY STUDENTS CHOOSE ALGORITHMIC THINKING AT MACEWAN UNIVERSITY

Upon completion, students will leave with a stronger foundation for future Computing Science studies, with the following outcomes:

- **Algorithm-first foundation:** step-by-step reasoning, debugging and structured problem solving.
- **Discipline and confidence:** learning to persist, test, and improve solutions.
- **Portfolio value:** Programming is aligned with professional business and industry standards for career-enhancing experiences. Students build a final project that can support future applications,
- **Future-ready pathways:** hands-on use with Python, commonly used in AI, machine learning, data science and engineering.
- **Global learning experience:** experience studying at a Canadian university campus, practicing English and communication in an academic environment.



LEARNING OUTCOMES

BY THE END OF ALGORITHMIC THINKING, STUDENTS WILL BE ABLE TO:

- Explain basic algorithmic ideas: sequencing, decisions, repetition (loops), and decomposition (functions).
- Write and debug beginner Python programs using variables, conditionals, loops, and functions.
- Use input/output and simple data structures to build interactive programs.
- Create a small portfolio project and present how it works with clear communication.

```
.WIDTH = 800
.HEIGHT = 800

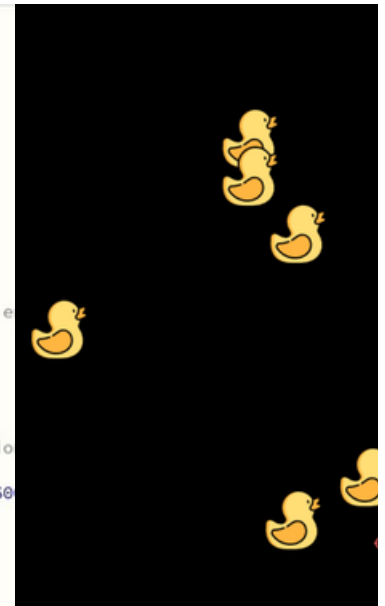
crosshair = Actor('crosshair')

# creating a new enemy
def newEnemy(pos):
    e = Actor('enemy', pos=pos)
    e.hit = False
    e.timer = 10
    e.hits = []
    return e

# creating a bullet that has hit an enemy
def newHit(pos):
    h = Actor('bullet', pos=pos)
    return h

#Continue PART 2.
# create 3 enemies at various positions
enemies = []
for p in [(0,200),(-200,400),(-400,600)]:
    enemies.append(newEnemy(p))

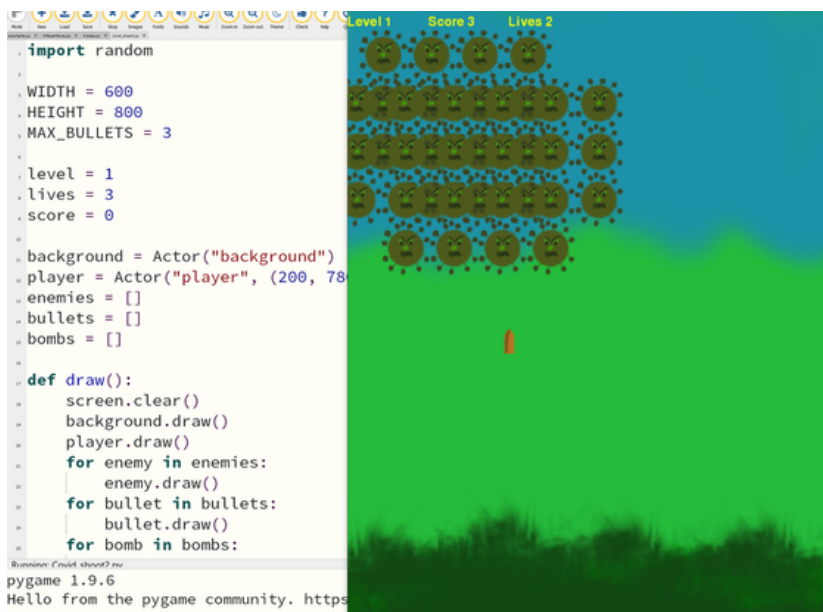
numberofbullets = 10
MAXBULLETS = 10
```



SAMPLE COURSE SCHEDULE

DAY	DAILY TOPIC FOCUS	DAILY LEARNING OUTCOMES
DAY 1	• Environment setup (install / verify Python & Mu IDE) • Explore interface and learn basic commands • Hands-on practice modifying first program (extra prints, colors) • Debugging basics (common errors, tracebacks)	• Students can run Python code using Mu Editor. • Students understand arithmetic operators and rounding. • Students can declare variables and use input() to collect user data. • Students recognize basic Python data types. • Students build confidence writing and modifying small scripts
DAY 2	• Variables and data types • Input basic math • Strings in depth (concept, slicing, methods) • Review & Q&A	• Students can create and use variables with appropriate data types. • Students can take user input and perform computations using type conversion. • Students can format output using f-strings and apply basic string operations. • Students complete a small creative text-based project.
DAY 3	• Conditionals: logical and comparison operators • Live demo • Mini project • Showcase & feedback sessions	• Students can write conditional logic using if/elif/else. • Students can build simple menu-based programs with basic validation. • Students complete a playable mini-project and explain its logic.
DAY 4	• Loops (for, while, break, continue) • Quiz 1 (vars, strings, decisions, loops) • Loop challenge lab	• Students can use for and while loops for repetition and counting. • Students can combine loops with conditionals and input. • Students complete loop-based challenges and identify common loop bugs.
DAY 5	• Loop challenge continues (problem solving & pseudocode) • Debugging strategies • Week 1 key takeaways and wrap-ups • Open lab & catch up	• Students can approach coding problems using pseudocode and step-by-step reasoning. • Students can debug using print tracing and basic test cases. • Students consolidate Week 1 skills and complete any missing work.
DAY 6	• Lists & Dictionaries • Practice (building small lists + dict trackers) • Quiz 2 • Applying data structures to games	• Students can store and update data using lists and dictionaries. • Students can choose an appropriate data structure for a simple game need. • Students complete Quiz 2 and identify next learning goals.
DAY 7	• Functions (design & reuse) • Parameters, return values, scope • Refractor messy script into clean functions • Break game logic	• Students can create and use functions with parameters and return values. • Students can refactor repeated code into reusable helpers. • Students can structure simple game logic using functions.

Schedules can vary based on duration of educational programming.



ADMISSION REQUIREMENTS

- Participants must be twelve years of age or higher (NB - custodianship cannot be provided by MacEwan University.)
- Minimum English language proficiency of B1 or better in the Common European Framework of Reference for Languages (CEFR)
- Proof of travel insurance is required at time of registration

SAMPLE COURSE SCHEDULE

DAY	DAILY TOPIC FOCUS	DAILY LEARNING OUTCOMES
DAY 8	• Error handling • Debugging patterns and safe input techniques • Mini project planning (teams, solo) • Practice: robust input handling	• Students can validate input and handle errors using try/except. • Students can plan a small project with clear features and tasks. • Students create an initial prototype with robust input handling.
DAY 9	• Modules (import, using built ins, simple structure) • Turtle basics: drawing, movement, coordinates • Begin visual components (simple shapes, sprites and interaction)	• Students can import and use Python modules appropriately. • Students can create simple Turtle graphics with movement and basic interaction. • Students build a foundation for a small Turtle-based game.
DAY 10	• Turtle game lab (Guided Build) • Core game loop, keyboard input and controls • Intro to Krita (quick workflows) • Small prototyping based on learnings	• Students can build a small Turtle game with controls and scoring. • Students can create simple graphical assets in Krita and export them correctly. • Students integrate basic art assets into a working prototype.
DAY 11	• Audacity introduction • Recording audio, trim & basic effects • Exporting & saving as WAV/MP3 • Integrating sound in Python (playsound, pygame) • Mini-demo	• Students can create and export basic audio assets using Audacity. • Students can integrate sound effects into a Python project. • Students enhance an existing game with audio feedback.
DAY 12	• Advanced Turtle + optional Python extras • Animations, additional events • Game polishing • Final game project planning • Core mechanics, asset list and task breakdown	• Students apply advanced Turtle concepts to improve a game. • Students create a clear plan for the final project, including assets and roles. • Students set up a working starter structure for Game 3.
DAY 13	• Game coding sprint • Core structure • Menus & game states • Asset integration • Instructor coaching and peer review	• Students implement a complete Game 3 structure with states and assets. • Students improve reliability through debugging and testing. • Students practice peer review and iterative improvement.
DAY 14	• Final wrap-up • Testing checklist, final bug fixes • Presentation rehearsal, project demos • Recap key lessons	• Students finalize and present a working project. • Students demonstrate confidence in explaining their code and design choices. • Students leave with next-step resources and a portfolio-ready artifact.



MacEwan
UNIVERSITY

FOR PARTICIPANTS, THIS PROGRAM INCLUDES

- 3 meals per day
- Accommodation
- Health Insurance
- Transportation to / from Edmonton International Airport
- Transportation to / from campus (may be either a transit pass or shuttle bus, depending upon location of accommodation)

**Minimum enrollment required*



```
.WIDTH = 800
.HEIGHT = 800

.crosshair = Actor('crosshair')

# creating a new enemy
def newEnemy(pos):
    e = Actor('enemy', pos=pos)
    e.hit = False
    e.timer = 10
    e.hits = []
    return e

# creating a bullet that has hit an enemy
def newHit(pos):
    h = Actor('bullet', pos=pos)
    return h
```

HOW TO APPLY

Start your adventure with MacEwan University today.

To plan for your group, email
SCEInternational@MacEwan.ca.

ABOUT MACEWAN UNIVERSITY

MacEwan University is the 3rd largest undergraduate university in Western Canada with a vibrant multicultural student community from all over the world. We are a student focused university that follows business and industry standards to align our programming to provide students with career enhancing experiences. We can build unique Educational Tour programs that incorporate sample lessons from any certificate, diploma or degree offered through MacEwan University.

EXPAND YOUR LEARNING

School of Continuing Education
Allard Hall, Room 11-521
11110 - 104 Avenue
Edmonton, AB T5K 1M9

+1-780-497-4400
sceinternational@macewan.ca
MacEwan.ca/sceinternational

Office hours
Monday to Friday 8:30 a.m. - 4:30 p.m. MST

Follow us @MacEwanSCE

