1. Computational Thinking (CT)
التفكير الحاسوبي (CT)

- Definition: An approach to problem-solving using principles from computer science.
التعريف: منهجية لحل المشكلات باستخدام مبادئ من علم الحاسوب.
- Key Aspects:
الجوانب الرئيسية:

  - Extracting essential details of a problem.
استخلاص التفاصيل الأساسية للمشكلة.
  - Formulating problems for computer understanding.
صياغة المشكلات بطريقة يمكن للحاسوب فهمها.
  - Automating problem-solving processes.
أتمتة عمليات حل المشكلات.
- Not Just for Programmers: Applicable in diverse fields (e.g., graduation ceremonies, climate prediction, legal assistance).
ليس للمبرمجين فقط: قابلة للتطبيق في مجالات متنوعة (مثل حفلات التخرج، التنبؤ المناخي، المساعدة القانونية).
- Goal: Empower individuals to solve problems efficiently with computers, not just to think like computer scientists.
الهدف: تمكين الأفراد من حل المشكلات بكفاءة باستخدام الحواسيب، وليس فقط التفكير مثل علماء الحاسوب.

---

2. Logical Thinking
التفكير المنطقي

- Boolean Logic:
المنطق البولياني:

  - Deals with true (1) or false (0) values.
يتعامل مع قيم صحيحة (1) أو خاطئة (0).
  - Propositions: Clear, unambiguous statements.
الافتراضيات: عبارات واضحة وغير غامضة.
  - Compound Propositions: Combined using logical operators.
الافتراضيات المركبة: يتم دمجها باستخدام العوامل المنطقية.
- Symbolic Logic: Uses symbols (e.g., P, Q, S) to represent statements.
المنطق الرمزي: يستخدم رموزاً (مثل P, Q, S) لتمثيل العبارات.
- Logical Operators:
العوامل المنطقية:

  - AND (∧), OR (∨), NOT (¬), IMPLIES (→), IF AND ONLY IF (↔).
AND (وَ), OR (أو), NOT (ليس), IMPLIES (يستلزم), IF AND ONLY IF (إذا وفقط إذا).
  - Truth tables define their behavior.
جداول الحقيقة تحدد سلوكها.

---

3. **Algorithmic Thinking**
   التفكير الخوارزمي

- **Algorithm: A finite sequence of unambiguous instructions with a clear start and end.**
  الخوارزمية: سلسلة محدودة من التعليمات الواضحة ذات بداية ونهاية محددة.
- **Representations:**
  تمثيلات:

- **Pseudocode: Informal step-by-step description.**
  الكود الزائف: وصف غير رسمي خطوة بخطوة.
- **Flowchart: Graphical representation using symbols and arrows.**
  مخطط الانسياب: تمثيل رسومي باستخدام رموز وأسهم.
- **Variables: Placeholders to store and update data during execution.**
  المتغيرات: عناصر نائبة لتخزين وتحديث البيانات أثناء التنفيذ.
- **Control Structures:**
  هياكل التحكم:

- **Selection (if-else): Decision-making based on conditions.**
  الاختيار (if-else): اتخاذ القرار بناءً على الشروط.
- **Iteration (looping): Repeating steps (e.g., for, while loops).**
  التكرار (الحلقات): تكرار الخطوات (مثل حلقات for, while).
- **Example: Game logic (e.g., noughts and crosses) using loops and conditionals.**
  مثال: منطق اللعبة (مثل X-O) باستخدام الحلقات والشروط.

---

4. **Common Pitfalls**
   المزالق الشائعة

- **Clarity and Meticulousness:**
  الوضوح والدقة:

- **Computers follow instructions literally (e.g., division by zero causes crashes).**
  الحواسيب تتبع التعليمات حرفياً (مثل: القسمة على الصفر تسبب تعطل النظام).
- **No innate intelligence or common sense.**
  لا تمتلك ذكاءً فطرياً أو حساً مشتركاً.
- **Logical Operator Misuse:**
  سوء استخدام العوامل المنطقية:

- **Computers interpret operators strictly (e.g., "A and B" vs. human interpretation).**
  الحواسيب تفسر العوامل بدقة (مثل: "A and B" مقابل التفسير البشري).
- **Missing Eventualities:**
  إغفال الاحتمالات:

- **Incomplete conditionals (e.g., ignoring edge cases like exact 50% scores).**
  شروط غير مكتملة (مثل: تجاهل الحالات الطرفية كالنتيجة 50% بالضبط).
- **Complex Conditionals:**
  الشروط المعقدة:

- **Use parentheses to clarify precedence (e.g., ¬(score < 51) V score > 80).**
  استخدم الأقواس لتوضيح الأولوية (مثل: ¬(score < 51) V score > 80).

- Order of Precedence:
  ترتيب الأولوية:
  - Mathematical: Parentheses > Exponents > Multiplication/Division > Addition/Subtraction.
    الرياضيات: الأقواس > الأسس > الضرب/القسمة > الجمع/الطرح.
  - Logical: Parentheses > NOT > Comparisons (>, <) > AND > OR.
    المنطق: الأقواس > NOT > المقارنات (>, <) > AND > OR.

---

5. Problem-Solving Systematic Approach
   المنهجية النظامية لحل المشكلات
1. Understand the Problem: Restate it, use diagrams, identify knowns/unknowns.
   فهم المشكلة: إعادة صياغتها، استخدام الرسوم البيانية، تحديد المعروفات والمجهولات.
2. Devise a Plan: Decomposition, recursion, tree structures.
   وضع خطة: التحليل، التكرار، الهياكل الشجرية.
3. Execute the Plan: Implement the solution.
   تنفيذ الخطة: تطبيق الحل.
4. Review and Extend: Evaluate results and generalize patterns.
   مراجعة وتوسيع: تقييم النتائج وتعميم الأنماط.
   - Heuristics: Strategies for good-enough solutions (not necessarily optimal).
     الإرشادات: استراتيجيات لحلول جيدة كافية (ليست بالضرورة مثالية).
   - Collaboration and Iteration: Key to quality problem-solving.
     التعاون والتكرار: مفتاح حل المشكلات بجودة عالية.

---

6. Decomposition and Patterns
   التحليل والأنماط
   - Decomposition:
     التحليل:
     - Break complex problems into simpler, independent sub-problems.
       تقسيم المشكلات المعقدة إلى مشكلات فرعية أبسط ومستقلة.
     - Facilitates collaboration and divide-and-conquer strategies.
       يسهل التعاون واستراتيجيات "فرق تَسُد".
   - Tree Structures: Hierarchical representation of tasks (e.g., science project breakdown).
     الهياكل الشجرية: تمثيل هرمي للمهام (مثل: تحليل مشروع علمي).
   - Generalization:
     التعميم:
     - Loops: For repeating patterns (e.g., drawing multiple eyes).
       الحلقات: للتكرار الأنماط (مثل: رسم عيون متعددة).
     - Subroutines: Reusable code blocks (e.g., draw_eye(r, x, y)).
       الإجراءات الفرعية: كتل برمجية قابلة لإعادة الاستخدام (مثل: draw_eye(r, x, y)).

- **Rules: Encoded relationships (e.g., inner circle radius = half of outer circle).**
  القواعد: علاقات مُرمَّزة (مثل: نصف قطر الدائرة الداخلية = نصف قطر الدائرة الخارجية).