| | |
|---|---|
| Module designation | **Computational Thinking**<br>**Indonesian title:** *Berpikir Komputasional*<br>**Course code:** 8420202239 |
| Semester(s) in which the module is taught | Semester 8 (Fourth year) |
| Person responsible for the module | **Coordinator:** Dr. Yurizka Melia Sari, M.Pd.<br>**Teaching staff:** Novita Vindri Harini, M.Pd. |
| Language | Indonesian |
| Relation to curriculum | Elective |
| Teaching methods | Lectures, unplugged and hands-on computing activities, problem-based learning, project-based learning, and group discussions<br>**Media and tools:** Slides, LCD projector, whiteboard, UNESA Learning Management System (LMS), unplugged activity materials, spreadsheet software (Excel or Google Sheets), Scratch, Python (Google Colab), GeoGebra |
| Workload (incl. contact hours, self-study hours) | **(Estimated) total workload:** 2 CU × 170 minutes × 16 weeks = 5,440 minutes = 90.67 hours (≈ 3.18 ECTS)<br>**Contact hours:** lectures, 2 × 50 minutes per week × 16 weeks = 26.67 hours (class hours per week: 340 minutes = 5.67 hours, including structured activities)<br>**Private study including examination preparation:** structured assignments, 2 × 60 minutes per week × 16 weeks = 32.00 hours; independent study, 2 × 60 minutes per week × 16 weeks = 32.00 hours (total 64.00 hours) |
| Credit points | 2 CU (3.18 ECTS)<br>*1 CU = 170 minutes per week × 16 weeks = 45.3 hours; 1 ECTS = 28.48 hours (144 CU = 229 ECTS), so 1 CU = 1.59 ECTS (UNESA Rector's Decree No. 736/HK/KM/2023).* |
| Required and recommended prerequisites for joining the module | None |
| Module objectives/intended learning outcomes | **Programme learning outcomes (PLO) addressed:**<br>• PLO-3: Develop logical, critical, systematic and creative thinking in carrying out specific work in the field of expertise and in accordance with the work competency standards in the relevant field.<br>• PLO-5: Apply the basic principles of mathematics to solve simple mathematical problems.<br>• PLO-6: Demonstrate mathematical knowledge and insight.<br>**Course learning outcomes (CLO): after completing this module, students are able to:**<br>• CLO-1: Explain computational thinking and its components (decomposition, pattern recognition, abstraction, algorithmic thinking, and evaluation) and relate them to mathematical problem solving (PLO 3)<br>• CLO-2: Apply decomposition, pattern recognition, and abstraction to solve basic mathematical problems on number patterns, sequences, geometry, and probability, and generalise the patterns found (PLO 5) |

<table>
<tr><td></td><td>
<ul>
<li>CLO-3: Design, represent (flowcharts and pseudocode), and verify algorithms for mathematical procedures, and analyse their correctness and efficiency (PLO 6)</li>
<li>CLO-4: Implement algorithms and simulations for mathematical problems using spreadsheets, a programming environment, and GeoGebra (PLO 5)</li>
<li>CLO-5: Test, debug, and evaluate computational solutions, and develop and present a computational solution to a mathematical problem together with a learning activity that uses computational thinking (PLO 3)</li>
</ul>
</td></tr>
<tr><td>Content</td><td>This course introduces computational thinking as a way of solving mathematical problems. Students learn its core components, namely decomposition, pattern recognition, abstraction, and algorithmic thinking, together with testing, debugging, and evaluation, and apply them to problems on number patterns, sequences, geometry, and probability. Students design and verify algorithms for mathematical procedures, represent them as flowcharts and pseudocode, and implement them with spreadsheets, a programming environment such as Scratch or Python, and GeoGebra. They also use simulations to explore mathematical ideas, and look at how computational thinking can be integrated and assessed in school mathematics, for example through Bebras-style tasks. Learning is problem-based and project-based, ending in a final project in which students develop a computational solution to a mathematical problem and a short learning activity that uses it.</td></tr>
<tr><td>Examination forms</td><td>Final score is calculated from the weekly assessment scheme below (CLO/PLO addressed and weight of each assessment):
<table>
<tr><th>Week</th><th>CLO</th><th>PLO</th><th>Weight (%)</th></tr>
<tr><td>1</td><td>CLO-1</td><td>PLO-3</td><td>3</td></tr>
<tr><td>2</td><td>CLO-1</td><td>PLO-3</td><td>3</td></tr>
<tr><td>3</td><td>CLO-2</td><td>PLO-5</td><td>3</td></tr>
<tr><td>4</td><td>CLO-2</td><td>PLO-5</td><td>3</td></tr>
<tr><td>5</td><td>CLO-2</td><td>PLO-5</td><td>3</td></tr>
<tr><td>6</td><td>CLO-3</td><td>PLO-6</td><td>3</td></tr>
<tr><td>7</td><td>CLO-3</td><td>PLO-6</td><td>3</td></tr>
<tr><td>8</td><td>CLO-3</td><td>PLO-6</td><td>20</td></tr>
<tr><td>9</td><td>CLO-4</td><td>PLO-5</td><td>4</td></tr>
<tr><td>10</td><td>CLO-4</td><td>PLO-5</td><td>4</td></tr>
<tr><td>11</td><td>CLO-4</td><td>PLO-5</td><td>4</td></tr>
<tr><td>12</td><td>CLO-4</td><td>PLO-5</td><td>4</td></tr>
<tr><td>13</td><td>CLO-5</td><td>PLO-3</td><td>4</td></tr>
<tr><td>14</td><td>CLO-5</td><td>PLO-3</td><td>4</td></tr>
<tr><td>15</td><td>CLO-5</td><td>PLO-3</td><td>5</td></tr>
<tr><td>16</td><td>CLO-5</td><td>PLO-3</td><td>30</td></tr>
</table>
</td></tr>
<tr><td>Study and examination requirements</td><td>Students are considered competent and pass if the final score, calculated from weekly tasks and participation (45%), the midterm computational problem solving portfolio (20%), the final project presentation (5%), and the final examination (30%), is at least 55 or C.</td></tr>
</table>

**Final index (grade conversion):**

| Index | Converted score | Score range (A = final score) |
|---|---|---|
| A | 4.00 | 85 ≤ A ≤ 100 |
| A- | 3.75 | 80 ≤ A < 85 |
| B+ | 3.50 | 75 ≤ A < 80 |
| B | 3.00 | 70 ≤ A < 75 |
| B- | 2.75 | 65 ≤ A < 70 |
| C+ | 2.50 | 60 ≤ A < 65 |
| C | 2.00 | 55 ≤ A < 60 |
| D | 1.00 | 40 ≤ A < 55 |
| E | 0.00 | 0 ≤ A < 40 |

Reading list


1. Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.
2. Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. Educational Researcher, 42(1), 38-43.
3. Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147.
4. Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational Research Review, 22, 142-158.
5. Polya, G. (1945). How to solve it. Princeton University Press.
6. Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Paper presented at the Annual Meeting of the American Educational Research Association, Vancouver, Canada.
7. Román-González, M., Pérez-González, J.-C., & Jiménez- Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Computers in Human Behavior, 72, 678-691.
8. Dagienė, V., & Stupuriene, G. (2016). Bebras: A sustainable community building model for the concept based learning of informatics and computational thinking. Informatics in Education, 15(1), 25-44.