iCOMPUTE Level 1
Parent FAQs

What is iCOMPUTE?

iCOMPUTE is program targeted at Ismaili youth. It aims to bolster participation in Science, Technology, Engineering, and Mathematics (STEM), specifically Computer Science and Software Engineering, by providing youth with exposure to the field and opportunities and support to build pertinent skills and develop a growth mindset.

iCOMPUTE consists of three levels. In Level 1, participants work with Scratch to learn basic computational concepts. In Level 2, participants apply computational concepts to build an app using Appinventor. In Level 3, participants are introduced to the basic principles of descriptive and inferential statistics using Python.

What is a growth mindset?

Growth mindset is an idea articulated by Carol Dweck, a Stanford psychologist. Individuals with a growth mindset believe that traits such as talent and intelligence can be developed by hard work and dedication[1]. The alternative to a growth mindset is a fixed mindset in which individuals believe that their traits are static[2]. Individuals with a growth mindset embrace effort and take setbacks well, while individuals with a fixed mindset eschew challenges and withdraw when faced with failure[3].

Dweck's research found that a growth mindset amongst students is associated with an upwards trajectory in academic performance while a fixed mindset is associated with flat trajectory[4]. Furthermore, an intervention that introduces students to a growth mindset can reverse a downwards trend in academic performance[5].

Is iCOMPUTE Level 1 right for my child?

iCOMPUTE Level 1 aims to teach computational thinking to children using Scratch, a block based programming language. In applying computational concepts, participants are exposed

---

[1] Dweck, C. (2006). Mindset: The New Psychology of Success. Random House: New York.
[2] Dweck, C. (2006). Mindset: The New Psychology of Success. Random House: New York.
[3] Dweck, C. (2006). Mindset: The New Psychology of Success. Random House: New York.
[4] Blackwell, L., Trzesniewski, K., & Dweck, C. (2007). Implicit Theories of Intelligence Predict Achievement Across an Adolescent Transition: A Longitudinal Study and an Intervention. Child Development, Vol. 78, No. 1, pp. 246-263.
[5] Blackwell, L., Trzesniewski, K., & Dweck, C. (2007). Implicit Theories of Intelligence Predict Achievement Across an Adolescent Transition: A Longitudinal Study and an Intervention. Child Development, Vol. 78, No. 1, pp. 246-263.

to and expected to follow the engineering process. The program is designed for students between the ages of 9 and 11. No prior experience is necessary.

What is computational thinking?

> "Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent"[6]

Why is computational thinking important?

In her article "Computational Thinking", Jeannette M. Wing the head of the Computer Science Department at Carnegie Mellon University argues: "Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability."[7]

In justifying why, she outlines many of the skills that computational thinking encompasses, skills that are transferable to almost all domains.[8] Such skills include problem solving, designing systems, thinking recursively, using abstraction and decomposition, using heuristic reasoning and thinking in terms of prevention, protection and recovery from worst-case scenarios.[9]

What will my child learn in iCOMPUTE LEVEL 1?

The curriculum teaches computational thinking by focusing on seven computational concepts that were identified by Karen Resnick and Mark Resnick in their article, "New frameworks for studying and assessing the development of computational thinking". Karen and Mark Resnick are part of the MIT Media lab team that developed Scratch.

These concepts are: sequences, loops, events, parallelism, conditionals, operators and data. The concepts identified are those useful in Scratch, and other programming and non-programming contexts[10].

The curriculum is divided into four sections. The first section contains 3 lessons designed to provide requisite background information and context. Subsequent sections include between 2 and 5 lessons, followed by a project. Each lesson teaches participants one or more computational concepts, and provides participants an opportunity to apply the concept both unplugged and within the Scratch framework. Each section terminates with a project.

---

[6] Jan Cuny, Larry Snyder, and Jeannette M. Wing, "Demystifying Computational Thinking for Non-Computer Scientists," work in progress, 2010.
[7] Wing, J.M., *Computational thinking.* Commun. ACM, 2006. **49**(3): p. 33-35
[8] Wing, J.M., *Computational thinking.* Commun. ACM, 2006. **49**(3): p. 33-35
[9] Wing, J.M., *Computational thinking.* Commun. ACM, 2006. **49**(3): p. 33-35
[10] Brennen, K. and M. Resnick, New Frameworks for Studying and Assessing the Development of Computational Thinking, in *Annual Meeting of the American Educational Research Association*. 2012: Vancouver, Canada

Projects provide participants with an opportunity to apply computational concepts explored in the all the lessons within the section as well as those introduced in former sections. Participants are encouraged to complete the projects using the engineering process.

How are projects organized?

Projects are created using Scratch and provide participants with an opportunity to practice applying computational concepts and exercise creativity. The steps required to complete each project lead participants through the engineering process. Participants are presented with a problem statement and design specifications, as well as examples of projects that meet these specifications. Participants are expected to brainstorm and design their solutions on paper prior to assembling code, and evaluate their solutions after assembling their code.

What is the schedule for iCOMPUTE Level 1?

Lessons and projects in this curriculum build on top of one another, and should be implemented in the order presented. A total of 21 hours is required to implement this curriculum in its entirety. To optimize learning, it is suggested that each session not exceed three (3) hours, and that no more than a week lapse between sessions. The outline in which the curriculum is implemented across seven days is presented below together with the topics covered by each lesson.

<table>
<tr><th></th><th>DAY 1</th><th>DAY 2</th><th>DAY 3</th><th>DAY 4</th><th>DAY 5</th><th>Day 6</th><th>Day 7</th></tr>
<tr><td>HOUR 1</td><td>Lesson 1</td><td>Lesson 6</td><td>Lesson 3</td><td>Project II</td><td>Lesson 11</td><td rowspan="3">Project III</td><td rowspan="3">Project III</td></tr>
<tr><td>HOUR 2</td><td>Lesson 4</td><td>Lesson 2</td><td>Lesson 7</td><td>Lesson 9</td><td>Lesson 12</td></tr>
<tr><td>HOUR 3</td><td>Lesson 5</td><td>Project I</td><td>Lesson 8</td><td>Lesson 10</td><td>Lesson 13</td></tr>
</table>

Unit One:

1. Lesson 1: Introduction to Computers
2. Lesson 2: Introduction to Engineering
3. Lesson 3: Growth Mindset

Unit Two:

1. Lesson 4: Introduction to the Scratch Framework
2. Lesson 5: Sequences
3. Lesson 6: Sequences & Iteration
4. Project One: Dream Sequences

Unit Three

1. Lesson 7: Parallelism
2. Lesson 8: Events
3. Project Two: Just for laughs

Unit Four

1. Lesson 9: Introduction to games
2. Lesson 10: Events in Games
3. Lesson 11: Conditionals
4. Lesson 12: Data
5. Lesson 13: Data & Operators
6. Project Three: Pac-Man

My child can only attend some weeks - can I still enroll him/her in iCOMPUTE Level 1?

iCOMPUTE Level 1 runs on a tight schedule; each session introduces new concepts that build on concepts from previous sessions. If your child misses a week, this will hinder his/her learning and will also create a challenge for facilitators. Please ensure that your child can attend all sessions prior to registering him/her.

What equipment/tools does my child need to attend iCOMPUTE Level 1?

Students must have access to a computer. There is both a web-based version of Scratch and a desktop version of Scratch for Mac and PC. To access the web-based version of Scratch, an internet connection and browser that supports flash is required. The Scratch platform can be accessed through the website: https://scratch.mit.edu/

Please note that standard browsers on tablets do not support flash and, consequently, cannot be used to access the web-based version of Scratch. Although the web browser Puffin can be used to access the web-based version of Scratch on tablet, the Scratch user interface is best suited to mouse and keyboard input rather than touch input.

We would like to ensure that all children are able to access the program, so if you have difficulty getting access to a computer or laptop, please contact your local AKEB Excellence in Education Member.

Students are expected to brainstorm and plan projects prior to assembling code. Consequently, students must also bring a notebook and writing tools to class.

I have a question, who can I contact?

If you have any questions, please contact the iCOMPUTE National Convener, Alysha Rahim at alysha.rahim@iicanada.net.

Last Revision: 27 August 2018