Livingstone Academy East London
An Aspirations Creative Technologies Academy

Aspirations Academies Trust
in association with Quaglia Institute

Draft 1.0 – Feedback to: office@livingstone-aspirations.org.

Livingstone Academies: **Problem-Solving and Computational Thinking Rubric**

Introduction

- This Problem-Solving and Computational Thinking Rubric has been developed through classroom observations, seeking advice from expert and experienced teachers, and researching the 21st Century Skills and computational thinking skills domain. This rubric is a first draft effort by a group of educators on articulating learning behaviours and dispositions of learners in the classroom.

- The learning behaviours and dispositions have not been grouped by age group or phase of education because there is insufficient academic research into this field. This document should be considered in context, that is, both age and ability appropriate, and should be applied to school curriculum and subject being taught. Teachers are encouraged to adapt the rubric based on their classroom experiences and share it with the wider community.

- The Problem-Solving and Computational Thinking Rubric is focused on learners and not the concepts underpinning problem solving, critical thinking, computational thinking or other overlapping thinking skills from other subject disciplines. Instead, these concepts are implicit in the statements. The learning behaviours in the 'realised' column may be aspirational for learners in primary (ages 5 – 11) and early secondary (ages 11-13) education. Teachers should apply professional judgement in how much to share and how to explain new or complex vocabulary to some learners.

- This Problem-solving and Computational Thinking Rubric can be a valuable tool for teachers when planning and delivering lessons ensuring learners have opportunities to develop positive attitudes to, for example, complexity or ambiguity. This is because the response of learners to learning opportunities (behaviourism) is dependent upon the activity, and both the way in which the activities are presented to learners by the teacher (constructivism and constructionism) and how learner processes the newly acquired knowledge (cognitivism). Therefore, there is a relationship between the 'approaches' by teachers and the behaviours exhibited by learners' and as such 'approaches' will feature in the rubric to provide some context.

- Teachers might also find the Problem-Solving and Computational Thinking Rubric useful in avoiding the use of passive verbs to describe learning outcomes (e.g. be aware of, be familiar with, know, understand, etc.) because the associated behaviours are often internal or not public, that is, they are not observable and therefore we can never really understand how the learner processes the newly acquired knowledge.

Created by: Mark Dorling and Thomas Stephens
Tested and edited by: Teachers form Aspirations Academies Trust, form the Digital Schoolhouse satellite schools, Phil Bagge, Andrew Csizmadia, Pete Marshman, Thomas Ng, Shahneila Saeed and Dr. Chris Stephenson
Feedback and review from the wider Initial Teacher Training and Computing Education community.

Livingstone Academy East London
An Aspirations Creative Technologies Academy

Aspirations Academies Trust
in association with Quaglia Institute

- This rubric should be regarded as a tool to aid (self, peer and teacher) assessment of learners Problem-Solving in any subject. Problem solving in any subject like any skill requires repeated exposure to a range of real world problems and how they are studied, solved and represented. Therefore, this should be used periodically in conjunction with a learner portfolio of work to make an overall judgement on learners Problem-solving and Computational Thinking capabilities.

- Teachers may find that the rubric can be used as an aid during lesson observations alongside an updated lesson observation form (see CAS QuickStart Computing reference below) containing computational thinking prompts because it can help mentors/coaches to consider if particular learning behaviours are or are not present in learners due to the design and delivery of teaching materials or learner attitudes and motivations.

Learning Attributes*	Hibernating	Passive	Active	Realised
"Confidence in understanding problems and dealing with complexity"	● Makes no effort to attempt to grasp any aspects of the problem. ● Avoids problems/challenges that have more than one step or part to solving them. ● May unintentionally overly complicate problems.	● Gathers some of the necessary information to be able to understand the problem. ● Grasps some but not all aspects of the problem often making educated guesses. ● Consistently seeks advice and reassurance. ● Requires support on how to approach each part of the problem/challenge.	● Gathers all necessary information to understand the problem. ● Understands all aspects of a problem. ● Displays independence in breaking down problems and filtering out unnecessary information. ● With support, builds solutions in parts (sub-solutions) to recompose for a final solution.	● Confidently and systematically breaks down problems and filters out unnecessary information and is able to explain the processes involved. ● Effectively organises all parts of a problem to develop coherent sub-solutions. ● Sub-solutions are appropriately coupled (joined). ● Reduces the complexity of the solution without affecting its behavior.
"Persistence in working with difficult problems"	● Doesn't engage with and avoids problems that are difficult to deal with and hard to solve.	● Reluctantly engages with difficult problems but doesn't persevere for long.	● Responds positively to difficult problems and validates outcomes. ● Displays persistence at times of difficulty.	● Relishes difficult problems and finds fun in being persistent. ● Able to explain how they have overcome difficult problems. ● Demonstrates resilience despite setbacks.

Created by: Mark Dorling and Thomas Stephens
Tested and edited by: Teachers form Aspirations Academies Trust, form the Digital Schoolhouse satellite schools, Phil Bagge, Andrew Csizmadia, Pete Marshman, Thomas Ng, Shahneila Saeed and Dr. Chris Stephenson
Feedback and review from the wider Initial Teacher Training and Computing Education community.

Livingstone Academy East London
An Aspirations Creative Technologies Academy

Aspirations Academies Trust
in association with Quaglia Institute

"Tolerance for ambiguity"	• Refuses to acknowledge or struggles to accept ambiguity in both problems/solutions. • Struggles to get started without a precise plan.	• Shows a willingness to acknowledge and accept some ambiguity exists in both problems/solutions.	• Shows an ability to tolerate ambiguity in both problems/solutions.	• Celebrates ambiguity and having different interpretations • Compares the performance of interpretations that do the same thing.
"Deals with open ended problems"	• Struggles to get started without a plan and clear expectations/deliverables. • Follows instructions and only does what they are told.	• Does minimum expected but no more. • Investigates a limited number of problems/solutions.	• Applies effort to independently explore an appropriate range of problems/solutions.	• Consistently displays curiosity to exhaustively investigate and analyse a broad range of appropriate problems/solutions.
"Adapts existing knowledge or solutions to solve new problems"	• Shows reluctance to or actively avoids learning from previous solutions.	• Struggles to identify patterns that match a problem to a previously learned solution. • When directed and with reassurance, will consider adapting pre-existing solutions to solve the current problem.	• Tinkers with solutions to find new uses. • With some support identifies patterns and trends in problems and solutions. • Chooses pre-existing solutions they are already aware of to adapt and solve the current problem.	• Independently identifies and acts on patterns in problems/solutions. • Independently seeks out pre-existing solutions (not directly within their existing sphere of knowledge or understanding) transferring ideas and/or solutions from one problem context to another.
"Iteratively develops, tests, and debug solutions"	• Struggles to express ideas as a solution. • Looks to submit the first working solution as the finished product. • Lacks an awareness of the need to test, debug and refine solutions iteratively. • Has a preferred way of representing solutions and often chooses this method regardless of the task.	• Implements a solution using a given (completed) design. • Prototypes solutions quickly and submits more than one iteration of a solution for feedback. • Begins to use logical reasoning to predict outcomes. • Shows an awareness of the need to debug solutions but requires constant support and advice during this process. • When directed will consider different ways to represent solutions.	• With support designs and models solutions. • Requires occasional support when testing and debugging solutions. • Uses logical reasoning to predict outcomes showing an awareness of inputs. • With support chooses an appropriate way to represent solutions. • Carefully records the iterative process and makes appropriate refinements to the solution.	• Independently designs, models, tests, debugs and refines solutions (using a test plan and data where appropriate). • Independently chooses an appropriate way to represent solutions. • Tidies up their solution to increase the comprehension of how it works. • Embeds comment and explanation of how a solution works to improve understandability and maintainability.

Created by: Mark Dorling and Thomas Stephens
Tested and edited by: Teachers form Aspirations Academies Trust, form the Digital Schoolhouse satellite schools, Phil Bagge, Andrew Csizmadia, Pete Marshman, Thomas Ng, Shahneila Saeed and Dr. Chris Stephenson
Feedback and review from the wider Initial Teacher Training and Computing Education community.

"Evaluates and weighs-up outcomes carefully"	● Doesn't consider evaluating the efficacy of a solution.	● Shows an awareness of the need to evaluate solutions against criteria but requires support and advice during this process.	● Independently evaluates the quality of solutions against given criteria. ● Considers if a solution is 'fit for purpose'. ● Views the work of others and identifies transferable efficiencies.	● Considers when good is good enough and uses this to design a criteria to critically evaluate the quality of a solution. ● Measures the efficiency of solutions to improve it. ● Makes trade-offs between conflicting demands then makes refinements and future solutions based on prioritising.
"Communicates and works with others to achieve a common goal or solution"	● Has a negative impact on others. ● Communicates ineffectively ● Makes inappropriate contributions to the group. ● Uses incorrect subject vocabulary. ● Has no sense of own strengths and weaknesses or those of other.	● Does not have a negative impact on others. ● Passively participates in the group, making no significant contribution. ● Uses the correct subject vocabulary. ● With prompts can explain how a solution works to others. ● Has a sense of own strengths and weaknesses but evidences little strategy to deal with them.	● Has a positive impact on the group. ● Communicates effectively with others. ● Groups working (with support) in parallel on the same problem/solution. ● Makes positive contributions to and supports others. ● Explains how a solution works to others. ● Understands own strengths and weaknesses and solicits help from appropriate others.	● Makes consistently positive contributions to the group. ● Groups working effectively in parallel on the same problem/solution. ● Balances autonomy and collaboration. ● Values others learning and teamwork styles to encourage contributions from others. ● Where appropriate, constructively leads others. ● Explains how a solution works with clarity. ● Uses team to effectively compensate for own weaknesses and uses own strengths to support others.

* Column a used inspiration for, and adapted from, the CTSA Operational definition of Computational Thinking for K12 Education

Created by: Mark Dorling and Thomas Stephens
Tested and edited by: Teachers form Aspirations Academies Trust, form the Digital Schoolhouse satellite schools, Phil Bagge, Andrew Csizmadia, Pete Marshman, Thomas Ng, Shahneila Saeed and Dr. Chris Stephenson
Feedback and review from the wider Initial Teacher Training and Computing Education community.

Bibliography

BBC Bitesized (2016), *Computational Thinking*
http://www.bbc.co.uk/education/topics/z7tp34j

Barr, V. & Stephenson, C. (2011), *Bringing Computational Thinking to K12*
http://www.amanyadav.org/CEP991A/wp-content/uploads/2014/08/Barr_Stephenson_2011.pdf

Bhattacharya, N., (2016), *Tinkering (an introduction)*
https://vimeo.com/182482442

Brennan, K., & Resnick, M. (2012), *New frameworks for studying and assessing the development of computational thinking*
http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Computing At School (2014), *CAS Barefoot Computational Thinking*
http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/

Csizmadia, A., Cuzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C. and Woollard, J. (2015), *Computational thinking: A guide for teachers*
https://community.computingatschool.org.uk/resources/2324

Computing At School (2015), *Lesson observation form with prompts*
http://www.quickstartcomputing.org/secondary/section4.html

CSTA (2011), *Operational definition of Computational Thinking for K12 Education*
https://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf

CSTA Voice (2016), *Irene Lee: Reclaiming the Roots of CT, Page 4*
https://csta.acm.org/Communications/sub/CSTAVoice_Files/csta_voice_03_2016.pdf

Department for Education (2014), *Computing Programmes of Study*
https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

Created by: Mark Dorling and Thomas Stephens
Tested and edited by: Teachers form Aspirations Academies Trust, form the Digital Schoolhouse satellite schools, Phil Bagge, Andrew Csizmadia, Pete Marshman, Thomas Ng, Shahneila Saeed and Dr. Chris Stephenson
Feedback and review from the wider Initial Teacher Training and Computing Education community.

Dorling, M. & Walker, M. (2014), *Computing At School Progression Pathways*
https://community.computingatschool.org.uk/resources/1692

Fuller et. Al. (2007), *Developing a computer science-specific learning taxonomy*
http://dl.acm.org/citation.cfm?id=1345438

Google (2016), *Computational Thinking for Educators*
https://computationalthinkingcourse.withgoogle.com/unit

Google (2016), *Computational Thinking from a Dispositions Perspective*
http://googleforeducation.blogspot.co.uk/2016/09/Computational-Thinking.html?m=1

Google (2016), *Exploring Computational Thinking*
https://www.google.com/edu/resources/programs/exploring-computational-thinking/index.html#!ct-overview

Potter. M, and Kustra (2012), *A primer on learning outcomes and the SOLO Taxonmy*
http://www1.uwindsor.ca/ctl/system/files/PRIMER-on-Learning-Outcomes.pdf

A focus on writing learning outcomes
https://afocusonlearningoutcomes.wordpress.com/resources/extras/

STEM Learning, *Computational Thinking*
https://www.stem.org.uk/elibrary/resource/35192

Created by: Mark Dorling and Thomas Stephens
Tested and edited by: Teachers form Aspirations Academies Trust, form the Digital Schoolhouse satellite schools, Phil Bagge, Andrew Csizmadia, Pete Marshman, Thomas Ng, Shahneila Saeed and Dr. Chris Stephenson
Feedback and review from the wider Initial Teacher Training and Computing Education community.