Arizona Draft Computer Science Standards
Technical Review Document

Reviewer Name: ____Ashish Amresh_____

Introduction Section

As you conduct your review of the introduction, please consider the following questions.

A. Does the introduction provide sufficient information and guidance on how to read the standards?
B. Does the introduction provide sufficient information on how the standards are structured?
C. Is there anything missing that should be included in the introduction?

Please provide feedback on the Introduction section. Include strengths and well as suggestions for refinements.

The introduction seems to emphasize the difference between computer technology standards and computer science standards. It would serve well if the focus was on what the standards are, what they mean for AZ schools, how they differentiate from existing frameworks and how would teachers/schools go about using these for adoption.

The message on equity in CS is lacking or weak. It seems to me that it has been added without significant thought on why it is important and how it should be addressed in k- 12. I would recommend the following resources on equity as good starting points to address this into the framework. https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf and https://services.google.com/fh/files/misc/diversity-gaps-in-computer-science-report.pdf

Standards Section by Grade Level

As you conduct your review of the grade level and High School standards, please consider these questions.

A. Does the introductory information for the grade band and for each grade level provide enough context to understand how the standards connect within the grade and between grades within each band?
B. Does each standard clearly state what students should know and be able to do?
C. Can the standard be measured?
D. Are there any ambiguous or unclear words/phrases?
E. Do the standards in each section have appropriate **breadth**?
F. Do the standards in each section have appropriate **depth of content and rigor** for the grade level?

G. Is there meaningful alignment and development of skills/knowledge within each grade and from one grade band/grade level to the next?

1. Please provide feedback on Kindergarten:

The rigor in Kindergarten and to some extent all elementary grades seems to be high at least from the point of view of the examples provided for the essential and sub concepts. It is my recommendation that lower grades, especially kindergarten should look at computing from experiences that children have in their daily life and connect with those experiences and how computing plays a role in them. Trying to provide technical depth in these grades runs the risk of alienating kids who otherwise would have embraced computing. Here is a good example of introducing coding without it being too technical. https://www.nais.org/learn/independent-ideas/november-2017/why-we-teach-coding-in-elementary-school/ (this is not an example for kindergarten, I am trying to give an example for ideas on how to teach CS in the lower grades)

It is also not necessary that all essential concepts be covered in each grade. For example, in Kindergarten it may be best to only cover impacts of computing and some subtopics from networks and the internet.

I recommend the following standards should be moved to higher grades or reworked with simpler examples: K.CS.T.1, K.NI.C.1, K.DA.IM.1, K.AP.M.1, K.AP.PD.3, K.IC.SLE.1

2. Please provide feedback on Grade 1:

First grade might be a good point to introduce the concept of what devices are and how they differ from one another. However, it is still premature to talk about trouble shooting hardware errors or describing the hardware layers. Same with cybersecurity and secured passwords, it is too early to talk about this in this grade. It might be good to introduce some data analysis from how to read charts and diagrams in this grade but data patterns and deriving meaning from these is out of scope.

I recommend 1.CS.T.1, 1.CS.T.2, 1.NI.C.1, 1.DA.IM.1, 1.AP.M.1, 1.AP.PD.3, 1.IC.SLE.1 be moved or reworked.

3. Please provide feedback on Grade 2:

Following with the above comments on changing the rigor, it would be good to introduce in second grade what secure computing is and why it is important. It would also be good to

introduce the idea of computation and how an algorithmic task provides solution by performing computations. I would recommend keeping the instruction at a very non-technical level, perhaps the use paper and pencil over actual devices would be preferred. It would also be important to have the standards allow for traditional methods of teaching without compromising on the technical content. An example of such an activity is to be found here: http://www.cse4k12.org/how_computers_work/index.html

I recommend 2.CS.T.2, 2.DA.S.1, 2.DA.IM.1, 2.AP.M.1, 2.AP.PD.3, 2.AP.PD.4 be moved or reworked

4. Please provide feedback on Grade 3:

The idea that students will connect globally is far-fetched for this grade. It would be good to introduce the idea of working together in teams in this grade to solve a common goal. However it is too early to expect the students to be ready to share/showcase their work with local/global audience. Instead it might be good to have in-class team presentations to promote the big idea of communication. This grade might be a good fit to bring in troubleshooting strategies. It is still early to introduce the concept of code fix and iterative program development, that should be brought in at a later grade. There are methods to teach computing with tools that identify and fix errors when they occur. This should be the strategy until this grade. For example the entire introduction to programming in Khan Academy uses this model.

I recommend 3.NI.NCO.1, 3.DA.IM.1, 3.AP.PD.2, 3.AP.PD.3, 3.AP.PD.5 be moved or reworked.

5. Please provide feedback on Grade 4:

Fourth grade would be a good place to start introducing the inner workings of the hardware and how each component interacts with one another. I have noticed that kids appreciate this better when they had a good grasp on how applications and programs work, so introducing this topic in this grade after they have had few years of experience writing simple programs will be a better strategy. Some of my previous grade move/rework suggestions on hardware and devices can now be brought in this grade. Comparing algorithms is not a trivial task, it requires some level of mathematical maturity and might not be a good idea for this grade. It might still be too early for code fixing and program iteration. This would be a good grade to introduce decomposition or divide and conquer strategies however, I believe it takes 2-3 years for students to get a good grasp on what the basic programming constructs are and how they operate. So, moving that to later grades will be beneficial.

I recommend 4.AP.M.2, 4.AP.PD.2, 4.AP.PD.3 be moved or reworked

6. Please provide feedback on Grade 5:

I would suggest that fifth grade be the first of the three focal points in the K-12 CS curriculum and the others being 8th grade and high school. At the end of this grade the student should be savvy enough to what a program is, how it works on the hardware, what the different devices mean and how they interact with each other, how technology impacts the world around us and

why it is essential for everyone to be skilled in CS. To this effect it might be beneficial use this grade as a review / revision of all the topics learnt to this point and not introduce any new topics. While the standards in the current form are building on top of each grade, it can sometimes lead to the impression that students coming from one grade to the other will know the content expected of them. It is more evident in CS than in other subjects that this is far from the truth, students need constant practice and review to be proficient with CS.

I recommend 5.NI.NCO.1, 5.DA.IM.1, 5.AP.A.1, 5.AP.M.2, 5.AP.PD.2, 5.AP.PD.3 be reworked or moved

7. Please provide feedback on Grade 6:

In sixth grade the ideas related to intellectual property, ethical conduct of code, piracy etc. can be introduced. This grade is a good place to introduce how to debug and fix code. It may also be a good place to introduce the concept of sharing code with others and looking at the idea of code repositories. I had avoided teaching networking big idea concepts till this grade. I believe that how information travels, how it is stored, how it is manipulated should be taught only after the student has had mature understanding of basic CS programming constructs and this grade would be suitable to introduce those topics. This grade is also suitable to compare different algorithms and explore the merits and demerits. It is still too early to talk about reliability and validity of programs. It is also early to talk about collecting requirements or getting feedback from end users.

I recommend 6.DA.S.1, 6.DA.IM.1, 6.NI.NCO.1 be moved or reworked

8. Please provide feedback on Grade 7:

This grade will be a good time to introduce the topics related to project management, scheduling, timelines, milestones. It will also be a good place to introduce the ideas of reliability, validity and usability (understanding the end-user perspective when deciding to create a program or application). The idea of binary notation and the concepts of bits and how that related to hardware should be introduced here. Networking topics that discuss various protocols can be taught in this grade. More practice with understanding the performance of algorithms, decomposition strategies, code sharing, debugging, documentation and communication should be emphasized.

9. Please provide feedback on Grade 8:

As per my 5th grade review, this grade is the second focal point in the CS curriculum and it would be appropriate to revise and review the topics and provide mechanisms to demonstrate the mastery of the topics learnt this far. To that effect, I recommend this grade be similar to an undergraduate capstone experience which provides a hands-on culminating point to showcase the CS skills. No new topic should be introduced, and the course is offered in a project-based setting where student teams are solving real world problems.

I recommend 8.NI.C.1, 8.NI.C.2, be moved.

10. Please provide feedback on High School:

High school is the appropriate place to really dig into cyber security. Here is a good framework to look at for suitable topics http://www.doe.nv.gov/uploadedFiles/ndedoenvgov/content/CTE/Programs/InfoMediaTech/Standards/Cybersecurity-STDS-ADA.PDF . I recommend that students should have introductory knowledge of threat detection, denial of service, spoofing, malware and ransomware. It is also important to introduce more hardware level concepts especially the idea that computers think in the binary notation and machine code is different from written code. It is important to introduce students to the idea of object-oriented programming when talking about abstraction. Students should have the programming skills to create classes and objects and manipulate them. Advanced topics such as inheritance or polymorphism may be avoided, however a discussion on what OOP is should not be missed. The goal for high school would be prepare the students to feel empowered in taking AP CSP and CSA classes and prior exposure to OOP helps improve student confidence. To this extent one of the culminating experiences (similar to 5th grade) would be for student teams to rework their programs using object orientation.

Standards Section organized by Essential Concept, Subconcept and learning progression

You have also been provided with each standard organized by essential concept to review and provide feedback on the development of the learning progression for each essential concept. As you conduct your review of the progression, please consider the following questions.

A. Does the standard address meaningful content within each essential concept?
B. Do the standards within each progression, including subconcepts have appropriate **depth of content and rigor**?
C. Is there meaningful alignment and development of skills/knowledge within each grade and from one grade level to the next for each progression?

My comment is broadly addressing all the 5 topics below. The standards are comprehensive in addressing all the essential concepts and have enough depth. The subconcepts are appropriate and cover the breath of topics for each essential concept. As per my previous comments there has been more depth in some grades than necessary and I have addressed those in the review. I have in my comments suggested the changes to alignment and progression.

1. **Please provide feedback on the Essential Concept <u>Computing Systems</u>:**

2. **Please provide feedback on the Essential Concept <u>Networking and the Internet</u>:**

3. **Please provide feedback on the Essential Concept Data and Analysis:**

4. **Please provide feedback on the Essential Concept Algorithms and Programming:**

5. **Please provide feedback on the Essential Concept Impacts of Computing:**

Glossary and Additional Resources

Please provide any additional comments related to the Glossary and Additional Resources section that you would like the working group to consider.

None

Additional Feedback

Please provide any additional comments about this draft that you want the revision working group to consider. Also use this an opportunity to summarize the strengths of the draft standards.

This draft does an excellent job of creating the breadth and depth necessary to teach CS in the K-1 setting. The chosen subtopics are relevant and pertinent to current needs for students to have CS skills. My two major concerns are 1) the standards seem to impose great technical rigor in the lower grades and 2) They do not provide opportunities for frequent review and reflection of the topics learnt. I have provided ideas and recommendations to address these.