AP® Computer Science Principles

Course Planning and Pacing Guide

Art Lopez
Sweetwater High School ▶ National City, CA

About the College Board

The College Board is a mission-driven not-for-profit organization that connects students to college success and opportunity. Founded in 1900, the College Board was created to expand access to higher education. Today, the membership association is made up of over 6,000 of the world's leading educational institutions and is dedicated to promoting excellence and equity in education. Each year, the College Board helps more than seven million students prepare for a successful transition to college through programs and services in college readiness and college success — including the SAT® and the Advanced Placement Program®. The organization also serves the education community through research and advocacy on behalf of students, educators, and schools. For further information, visit www.collegeboard.org.

AP® Equity and Access Policy

The College Board strongly encourages educators to make equitable access a guiding principle for their AP programs by giving all willing and academically prepared students the opportunity to participate in AP. We encourage the elimination of barriers that restrict access to AP for students from ethnic, racial, and socioeconomic groups that have been traditionally underrepresented. Schools should make every effort to ensure their AP classes reflect the diversity of their student population. The College Board also believes that all students should have access to academically challenging course work before they enroll in AP classes, which can prepare them for AP success. It is only through a commitment to equitable preparation and access that true equity and excellence can be achieved.

Welcome to the AP Computer Science Principles Course Planning and Pacing Guides

This guide is one of several course planning and pacing guides designed for AP® Computer Science Principles teachers. Each provides an example of how to design instruction for the AP course based on the author's teaching context (e.g., demographics, schedule, school type, setting). These course planning and pacing guides highlight how the components of the *AP Computer Science Principles Curriculum Framework* — including the learning objectives, essential knowledge statements, and computational thinking practices — are addressed in the course. Each guide also provides valuable suggestions for teaching the course, including the selection of resources, instructional activities, and classroom assessments. The authors have offered insight into the *why* and *how* behind their instructional choices — displayed along the right side of the individual unit plans — to aid in course planning for AP Computer Science Principles teachers.

The primary purpose of these comprehensive guides is to model approaches for planning and pacing curriculum throughout the school year. However, they can also help with syllabus development when used in conjunction with the resources created to support the AP Course Audit: the Syllabus Development Guide and the four Annotated Sample Syllabi. These resources include samples of evidence and illustrate a variety of strategies for meeting curricular requirements.

Contents

Instructional Setting

Sweetwater High School ▶ National City, CA

School

Sweetwater High School is a public high school located in National City, California — an urban and ethnically diverse city south of San Diego. The average class size is 30 students, and total enrollment is approximately 2,750. The school offers 51 AP courses.

Student population

The student population is:

- 80 percent Latino
- 14 percent Asian-American
- 3 percent other races not Hispanic
- 1.5 percent African-American
- 1 percent Caucasian
- 0.5 percent Native American or Alaskan Native

Free or reduced-price lunch is received by 89 percent of the students, 64 percent of the students are identified as English Language Learners, and 13 percent of the students have disabilities. Approximately 21 percent of students go on to four-year colleges, and approximately 60 percent go on to two-year programs. About three percent of students in the school take computer science courses, and one percent of those are female.

Instructional time

The school year begins at the end of July, and we have 114 instructional days. There are approximately 102 instructional days before the pilot AP Computer Science Principles Exam. Class meets three times a week. On two days, there is an approximately two-hour block. On one day, there is a 35–55 minute class, depending on professional learning communities, assemblies, and other such events.

Student preparation

The AP Computer Science Principles course is open to students in grades 9–12. The students vary widely in terms of prior computer science preparation, skills, and abilities: this is the first computer science course for the majority of the students. Most of the students are in 12th grade, with the ninth graders typically having the fewest participants in the course.

Instructional Setting (continued)

Primary planning resources

Expeditions Through Alice. Quintin Cutts, Sarah Esper, and Beth Simon. Accessed November 5, 2014. **https://sites.google.com/a/eng.ucsd.edu/expeditions-through-alice/home**.

This online book is the primary resource for the AP Computer Science Principles curriculum implemented at Sweetwater High School. It is separated into modules, based on the Alice software, that align to the curriculum framework. Alice is designed to help students learn computer science concepts via 3-D objects that are used to create short animations and video games.

Guides Through Alice 2. Quintin Cutts, Sarah Esper, and Beth Simon. Accessed November 5, 2014. **https://sites.google.com/a/eng.ucsd.edu/guides-through-alice-2/1-telling-a-story/book?pli=1**.

This companion website for the Expeditions Through Alice online book contains modules, videos, answers to the questions within the modules, lecture presentations that use a classroom-response system, and discussion questions to help students reach a deeper understanding of the learning objectives. It also contains labs with instructions and practice programming problems. In order to access the companion teacher's guide and instructional materials, please contact CS-CAVE (Creating a Village of CS Educators) at **http://cs-cave.org/index.php***, or Art Lopez at* **arthur.lopez@sweetwaterschools.org** *or* **mr.arturo.lopez@gmail.com***. Or to request access to all resource materials for Dr. Simon's curriculum go to* **http://www.ce21sandiego.org/***.*

Computer Science Principles Fall 2015 and Computer Science Principles Spring 2015. Arthur Lopez. Canvas. Accessed July 2, 2015. **https://canvas.instructure.com/courses/944870** (Fall Semester); **https://canvas.instructure.com/courses/943888** (Spring Semester).

I created and designed these online courses for my course based on Dr. Beth Simon's Alice-based Computer Science Principles course. Dr. Simon's course is taught at the University of California, San Diego, so I have modified some of the resources for a high school audience. The digital platform used for these courses is Canvas. ***Important note: the courses have been made public and direct URLs to assignments included with each activity. However, if any URL does not take you to the specific assignment, from the course homepage select Assignments on the far left and then search for the assignment title (which is always included in quotation marks in the activity's Materials)****.*

Alice (Version 2.4) [Software]. Carnegie Mellon University, 1999. Accessed November 4, 2014. **http://www.alice.org/index.php?page=downloads/download_alice2.4**.

"Curriculum." Exploring Computer Science. Joanna Goode and Gail Chapman. Accessed November 4, 2014. **http://www.exploringcs.org/curriculum**.

Instructional Setting (continued)

Primary planning resources

“CSTA - Curriculum Resources.” Computer Science Teachers Association/Association for Computing Machinery (ACM). Accessed December 12, 2014. **http://csta.acm.org/Curriculum/sub/CurrResources.html**.

“Pair Programming-in-a-Box: The Power of Collaborative Learning.” National Center for Women & Information Technology. Accessed November 7, 2014. **https://www.ncwit.org/pair-programming**.

Computer Science Unplugged. Michael Fellows, Tim Bell, and Ian Witten. Accessed November 8, 2014. **http://csunplugged.org/activities/**.

Computer Science Unplugged is a collection of free learning activities that teach computer science through engaging games and puzzles, using cards, string, crayons, and lots of running around.

“Repository for Alice Materials.” Susan Rodger. Duke University Department of Computer Science. Accessed November 7, 2014. **http://www.cs.duke.edu/csed/alice09/**.

Abelson, Hal, Ken Ledeen, and Harry Lewis. *Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion*. Addison-Wesley, 2008. Accessed November 7, 2014. **http://www.bitsbook.com/thebook/**.

The authors of this book describe how information is captured, processed, and made accessible, revealing how the digital revolution is changing the world in unexpected ways.

Dann, Wanda P., Stephen Cooper, and Randy Pausch. *Learning to Program with Alice: Custom Edition for University of California, San Diego*. Boston: Prentice Hall, 2009.

This book is from the Alice creators and is designed to teach students how to program with the platform.

Overview of the Course

I teach the AP Computer Science Principles course modeled after the curriculum designed by Dr. Beth Simon (University of California, San Diego). I use the programming platform Alice to teach students to create animated short stories, simulations, and video games by manipulating 3-D objects in a virtual world. Alice uses blocks to represent commands such as "move" or "turn" to make it easier for students to understand the fundamentals of programs. The programming environment avoids common syntax errors that frustrate many students in an introductory computer science course. Students taking the course are highly engaged and enjoy mastering computer science concepts.

There are a variety of assessment tools that are utilized during the course to continually assess and evaluate student performance and mastery. The students use an online book authored by Dr. Simon that contains exploratory modules based on the *AP Computer Science Principles Curriculum Framework*. Students are required to read, create exploratory "Alice world" projects, and answer a series of questions designed to provide a formative assessment and facilitate a deeper understanding of various concepts explored during the course.

Projects are planned, programmed, and tested to reinforce and demonstrate mastery of the learning objectives. The labs are designed to have students work independently or collaboratively, which research has shown to improve design, technical, and language skills, and increase the number of students who will take a more rigorous computer science course.

I use lecture presentations to scaffold each module and increase accessibility for all students. Each presentation builds upon students' prior knowledge, reinforcing concepts and skills, and poses essential questions based on the *AP Computer Science Principles Curriculum Framework*. Student engagement and participation is facilitated by the use of a classroom-response system (such as i>Clicker). At the beginning of each lecture, students take a low-stakes quiz based upon the exploratory module: this is an incentive for students to read the online book, complete exploratory projects, and provide feedback to me and their classmates. Peer-discussion questions, aided by the classroom-response system, are also included in the lecture presentation. These questions are designed to deepen students' understanding of computing concepts and help develop their technical, analytical, and communication skills through debate in teams that I moderate. I have taken the liberty of determining that the quiz questions are summative and the peer-discussion questions are formative. In addition to these assessments, a midterm and final are given as summative assessments for students to demonstrate mastery of required topics and concepts presented throughout the course.

Students are also required to complete a series of performance tasks based on the curriculum framework. These performance tasks include various computing activities such as exploring issues involving technology and society, which require students to use online discussion forums, Internet resource search and analysis tools, and online publishing platforms. Other performance tasks require students to demonstrate mastery of various programming concepts as outlined by the curriculum framework.

To broaden participation in the course, I initiate discussions with the administration, school counselors, teaching staff, and parents, and I regularly remind them of the course benefits, which include providing students a solid understanding of computing concepts and computational thinking skills. I recruit students from the local middle schools, the high school AVID program, and AP and general education courses — showing inspirational videos and presentations.

Computational Thinking Practices

P1: Connecting Computing

Developments in computing have far-reaching effects on society and have led to significant innovations. The developments have implications for individuals, society, commercial markets, and innovation. Students in this course study these effects, and they learn to draw connections between different computing concepts. Students are expected to:

- Identify impacts of computing.
- Describe connections between people and computing.
- Explain connections between computing concepts.

P2: Creating Computational Artifacts

Computing is a creative discipline in which creation takes many forms, such as remixing digital music, generating animations, developing websites and writing programs. Students in this course engage in the creative aspects of computing by designing and developing interesting computational artifacts as well as by applying computing techniques to creatively solve problems. Students are expected to:

- Create an artifact with a practical, personal, or societal intent.
- Select appropriate techniques to develop a computational artifact.
- Use appropriate algorithmic and information management principles.

P3: Abstracting

Computational thinking requires understanding and applying abstraction at multiple levels, such as privacy in social networking applications, logic gates and bits, and the human genome project. Students in this course use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity. Students are expected to:

- Explain how data, information, or knowledge is represented for computational use.
- Explain how abstractions are used in computation or modeling.
- Identify abstractions.
- Describe modeling in a computational context.

P4: Analyzing Problems and Artifacts

The results and artifacts of computation and the computational techniques and strategies that generate them can be understood both intrinsically for what they are as well as for what they produce. They can also be analyzed and evaluated by applying aesthetic, mathematical, pragmatic, and other criteria. Students in this course design and produce solutions, models, and artifacts, and they evaluate and analyze their own computational work as well as the computational work others have produced. Students are expected to:

- Evaluate a proposed solution to a problem.
- Locate and correct errors.
- Explain how an artifact functions.
- Justify appropriateness and correctness of a solution, model, or artifact.

Computational Thinking Practices (continued)

P5: Communicating

Students in this course describe computation and the impact of technology and computation, explain and justify the design and appropriateness of their computational choices, and analyze and describe both computational artifacts and the results or behaviors of such artifacts. Communication includes written and oral descriptions supported by graphs, visualizations, and computational analysis. Students are expected to:

- Explain the meaning of a result in context.
- Describe computation with accurate and precise language, notations, or visualizations.
- Summarize the purpose of a computational artifact.

P6: Collaborating

Innovation can occur when people work together or independently. People working collaboratively can often achieve more than individuals working alone. Learning to collaborate effectively includes drawing on diverse perspectives, skills, and the backgrounds of peers to address complex and open-ended problems. Students in this course collaborate on a number of activities, including investigation of questions using data sets and in the production of computational artifacts. Students are expected to:

- Collaborate with another student in solving a computational problem.
- Collaborate with another student in producing an artifact.
- Share the workload by providing individual contributions to an overall collaborative effort.
- Foster a constructive, collaborative climate by resolving conflicts and facilitating the contributions of a partner or team member.
- Exchange knowledge and feedback with a partner or team member.
- Review and revise their work as needed to create a high-quality artifact.

Pacing Overview

Unit	Hours of Instruction	Unit Summary
1: Introduction to Computer Science Principles	15	In this unit, students discover that computer programmers and developers learn new things by exploring, creating, and having a written plan before starting to explore. They learn the basics of computer programming by converting a plan for a simple animation into an Alice program — translating the plan into code one step at a time and testing the code by executing it after each new line. Some steps in the plan written as a single action will be broken down into multiple instructions or code; to ensure students understand what they are exploring they need to make predictions about what the code will do before executing it.
2: Telling and Dividing a Story (Methods)	15	Breaking up code into methods is one way computer scientists or programmers use abstraction to manage complexity in computing. In Unit 2, students learn that computers can do some very complex things and that humans need to use abstraction to help them manage thinking about the instructions they give the computer in their programs.
3: Behaving the Same vs. Differently (Parameters)	15	Students learn what parameters are and that parameters are specified in a method. They learn that by supplying different parameters or values, methods and programs will behave differently. Students also learn that methods can take any number of parameters. An activity on how computing has impacted innovations in other fields is given at the end of this unit, and students determine the beneficial and harmful effects of these innovations on people and society.
4: Get into the Story (Events)	10	Students learn to create interactive computer programs by using events. They learn to define, describe, and explain events and learn how to work with events to enable an audience (other computer users) to interact with their programs. Students learn that there are many different kinds of events, enabling their users to interact with their programs in different ways. They find out how code they write to respond to an event can in some situations be reused to handle many other events.
5: The Internet and Binary Numbers	12	Using a series of readings, articles, and activities based on the Computer Science Unplugged website and the Computer Science Teachers Association CSP Internet module, students explore and learn about binary numbers and how Internet infrastructure works.

Pacing Overview (continued)

Unit	Hours of Instruction	Unit Summary
6: Mathematical Expressions and Functions	15	Students learn how to calculate values based on sizes and locations of objects to allow them to create programs (Alice worlds) that react more realistically. Students learn to use a function (purple blocks or tiles) of an object to get information about its height, width, and depth. They learn to create mathematical expressions to calculate amounts they want to control (e.g. by moving or turning) so that the objects can interact realistically based on their size and location. Students create their own functions where they can put complex mathematical calculations to abstract or hide complex expression calculation.
7: Choosing Your Path (If Statements)	10	Students learn to introduce conditional behaviors at decision points into their programs (e.g. "I will do this, but only if a particular condition holds") by using If-else statements (green blocks or tiles). They learn that this allows them to make a decision between two different behaviors, based on the value of a function. They also learn that If-else statements evaluate functions via mathematical expressions or Boolean operators such as $<$, $\leq$, $>$, $\geq$, = (*equal*), and ≠ (*not equal*).
8: More Complex Control of Execution	15	Students learn two different ways to handle complex conditions under which they want to execute code: using compound Boolean expressions or "nesting" If-else statements (putting one inside another). Students create programs where execution is controlled (using If-else statements) by complex sets of conditions. One way students can express these complex conditions is by combining with the Boolean operator and and the Boolean operator or. Another way to express complex conditions is by using multiple If-else statements, sometimes one after another and sometimes with one nested inside part of another.
9: Doing Things Over and Over (Loops)	10	Students learn that one of the real powers of computing is to ask the computer to repeat the same instructions over and over again. Students learn two kinds of loops — control structures similar to If-else statements and doTogethers — and they learn to use counted loops and while loops. Students revisit the concept of abstraction in order to break down a complex task that they want to have repeated multiple times. They learn that while loops can be used when they do not have a specific number of times that they know (in advance) a set of instructions should be repeated; loops use Boolean expressions to determine when to keep executing the code (when the expression is true) or to stop and go on (when the expression is false).

Pacing Overview (continued)

Unit	Hours of Instruction	Unit Summary
10: Grouping Items Together (Lists)	10	Students learn that with computers, they can make lists of objects to have them all perform the same action without telling each object to do it. They learn that they can specify if they want objects to perform the actions together or one at a time. Students learn to introduce randomness so that results from a program can have more "natural" actions and that looping can be used with lists and randomness to create more interesting games.
11: Spreadsheets and Large Data Sets	12	Students learn that the concepts behind Alice programming can apply to many mainstream computer applications, including spreadsheets. Many of the basic functions and programming concepts, such as COUNTIF in spreadsheets, are similar to functions and If-statements in Alice. Students also learn to work with large data sets and to manipulate and manage large worksheets and tables.

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:
▶ Exploratory Module 1 Alice World ▶ Module 1 Discussion Questions ▶ Lab 1 Alice World Projects (3)

Guiding Questions

▶ **What are computing and computational thinking and why are they important to learn?** ▶ **What is computer programming?** ▶ **Why is creating a plan important for creating a program?** ▶ **Why do experienced programmers use prediction when creating and writing programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.3.1:** Use models and simulations to represent phenomena. [P3] **LO 4.1.1:** develop an algorithm for implementation in a program. [P2]	**Web** Expeditions Through Alice homepage, ("AppendixAFirstWorld.a2w," "AppendixA-DancingBee.a2w") Dann, Cooper, Pausch, "Appendix A: Getting Started"	**Instructional Activity: Complete Appendix Alice Worlds** This activity is designed to teach students the basics of using the Alice programming platform. I demonstrate to the students two example worlds embedded within the program — a movie-style animation and an interactive program. Students then learn to create a 3-D Alice world based on the step-by-step instructions in Appendix A, including setting the world up, saving it, and adding objects to it using built-in methods (instructions) among other basic programming elements. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** B; **1.3.1** D, E; **5.1.1** A-D
	Web "Pair Programming-in-a-Box: The Power of Collaborative Learning" (click the **Download** button for all of this activity's materials)	**Instructional Activity: Pair Programming** This activity is an introduction to the concept and techniques of pair programming. I give a lecture on pair programming and students watch a video and read an article on it. A description of pair programming, the benefits and problems, pair programming basics, the do's and don'ts, dealing with differences, and how pair programming affects their grade are covered. Students answer a series of questions on the article, video, and lecture. **Essential knowledge addressed: 1.2.2** B; **1.2.4** A-F

Pair programming is a collaborative learning method in which students program in pairs instead of individually. Due to the high number of English Language Learners (and students with special needs) in my class, working in pairs helps students learn programming concepts.

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:
▶ Exploratory Module 1 Alice World ▶ Module 1 Discussion Questions ▶ Lab 1 Alice World Projects (3)

Guiding Questions

▶ **What are computing and computational thinking and why are they important to learn?** ▶ **What is computer programming?** ▶ **Why is creating a plan important for creating a program?** ▶ **Why do experienced programmers use prediction when creating and writing programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 4.1.2: Express an algorithm in a language. [P5] **LO 4.2.4:** Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity, or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4]	**Web** "Alice Tutorials Summers 2008–2015": "The Essentials of Alice (Bunny)," "The Essentials of Alice (Kangaroo)," "Alice and Daisies," and "Positioning Objects": instruction files [ppt or pdf] and finished worlds [.a2w files] for all four tutorials Computer Science Principles Fall 2015, "Module 1 Questions Assignment"	**Instructional Activity: Four Alice Beginner Tutorials** Students begin to explore programming by creating four Alice worlds based on four Duke University tutorials. Students work in pairs using pair programming techniques. My online Canvas course provides students access to instructional materials and a repository of files and resources. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** B; **1.2.3** A-C; **1.2.4** A-D; **1.2.5** A-D; **1.3.1** E; **4.1.1** A, B; **4.1.2** A-C; **5.1.1** A-D; **5.1.3** C, D; **5.4.1** E, F, K
	Web Expeditions Through Alice, "1. Telling a Story" Guides Through Alice 2, "1. Telling a Story: Book" (video explanations for section 1.3-1.7) "Planning with Pseudocode"	**Instructional Activity: Exploratory Module 1 Alice World** Students complete module 1 from Expeditions Through Alice. They are directed to type a specific set of instructions to create a program using Alice to learn a basic level of programming. Students read and follow a written plan for a simple animation. They convert the plan into "pseudocode" and provide the computer instructions for the animation; they learn they can do instructions **in order** or do instructions **together**. Students work in pairs based on pair programming techniques. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.3.1** A-C; **4.1.1** A, B; **4.1.2** A-C; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H- J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.4.1** A, E, F, H-L

Pseudocode is a simple way to write programming code in English (or any other language). Pseudocode is not an actual programming language: it uses short phrases to write code for programs before you actually create it in a specific language. Here is a link to a video tutorial on **pseudocode** *by the Khan Academy.*

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:

▶ Exploratory Module 1 Alice World ▶ Module 1 Discussion Questions ▶ Lab 1 Alice World Projects (3)

Guiding Questions

▶ **What are computing and computational thinking and why are they important to learn?** ▶ **What is computer programming?** ▶ **Why is creating a plan important for creating a program?** ▶ **Why do experienced programmers use prediction when creating and writing programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 10-11) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.3.1, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.2.1, 5.4.1.**	**Web** Computer Science Principles Fall 2015, "Module 1 Questions Assignment" **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 1 Questions** As students read and follow the instructions for creating the exploratory module 1 Alice world, they are also required to answer a series of questions in the module that promote a deeper understanding of the concepts they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting. A step-by-step instructional document for this instructional activity is provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** A, B; **1.2.3** A-C; **4.1.2** B, C; **4.2.4** A, B; **5.1.2** B-.J; **5.4.1** E, F, H-J, L
	Web Expeditions Through Alice, "1. Telling a Story"	**Instructional Activity: Module 1 Final Programming Assignment** Students think of a short story or scene and create their own Alice world. They are required to select a world, place five objects into it, and use the **Do in order** and **Do together** blocks/tiles. Students storyboard and sketch out a scene (either using a piece of paper or a computer program). They include and describe what is going to happen, in what order, and in some detail — for example, where will a **Do together** be used? When the storyboard is complete, students build and set up the world by adding objects and placing them in the world. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **4.1.1** A, B; **4.1.2** A-C; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.4.1** A, E, F, H-L

A key point is telling students not to skim the reading. I also model for them what a good answer is versus a bad answer. A good answer is projected so that all students can see it then compare their answers; most students change their answers as they may have only written one or a few words and not completely answered the question.

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:

- Exploratory Module 1 Alice World
- Module 1 Discussion Questions
- Lab 1 Alice World Projects (3)

Guiding Questions

- **What are computing and computational thinking and why are they important to learn?**
- **What is computer programming?**
- **Why is creating a plan important for creating a program?**
- **Why do experienced programmers use prediction when creating and writing programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 10-11) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.3.1, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.2.1, 5.4.1.**	**Software** Classroom-Response System such as i>Clicker **Web** Guides Through Alice 2, "1. Telling a Story: Lecture" ("Module 1 Telling a Story.pptx" ; both Summative and Formative assessments are embedded in the presentation)	**Summative Assessment: Module 1 Quiz** Students answer three multiple-choice questions using a classroom-response system with a unique identifier for each student. Students are graded on correct responses. Students also answer four open-ended questions based on the essential questions. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-E; **1.2.2** A, **1.2.3** A-C; **1.2.5** A, B; **1.3.1** D,E; **4.1.1** A, B; **4.1.2** B, **4.1.2** C; **4.2.4** A, B; **5.1.1** A; **5.1.2** A, B, D; **5.4.1** E, F, H

I use i>Clicker as the student response system; any classroom-response system can be used.

Online programs such as Socrative.com are available, but i>Clicker works well for pedagogy of clicker quiz and discussion participation questions.

This summative assessment answers all of the guiding questions for this unit.

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:
▶ Exploratory Module 1 Alice World ▶ Module 1 Discussion Questions ▶ Lab 1 Alice World Projects (3)

Guiding Questions

▶ **What are computing and computational thinking and why are they important to learn?** ▶ **What is computer programming?** ▶ **Why is creating a plan important for creating a program?** ▶ **Why do experienced programmers use prediction when creating and writing programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
All of the learning objectives from the first activities in this unit (pages 10-11) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.3.1, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.2.1, 5.4.1.**	**Software** Classroom-Response System **Web** Guides through Alice 2, "1. Telling a Story: Lecture" (all files on this page) Guides Through Alice, "Discussion Videos" (all videos on this page)	**Formative Assessment: Module 1 Peer Discussion Questions** Students work in groups of three to answer multiple-choice questions using a classroom-response system. Some questions contain short video clips that explain code. First students read the question and individually select an answer (1 minute). Then they discuss their answers with their peers (1–2 minutes), clearly explaining why they chose that answer. After the discussion, students answer the question again — either the same answer or a new one based on their group discussion (1 minute). I ask for two or three volunteers to share their answers and group discussions with the class; they explain which answer they chose and why. I then share the correct answer and explain clearly why it's correct and why the other ones are not. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D, **1.2.2** A, B; **1.2.3** A-C; **1.2.5** A-D; **1.3.1** D, E; **4.1.1** A, B; **4.1.2** B, C; **4.2.4** A, B; **5.1.1** A; **5.1.2** A, B, D-F, H-L	◀ *Feedback is provided for every question; students and I discuss which answer is correct and which are incorrect. Students receive full credit by answering at least 90 percent of the questions — dependant on participation and analyzing the questions and answers for a deeper understanding of computing concepts and to develop technical, analysis, and communication skills (not on whether they got the answer right). I train students on having effective discussion groups by showing videos that demonstrate good versus bad discussions.*

UNIT 1: INTRODUCTION TO COMPUTER SCIENCE PRINCIPLES

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:

▶ Exploratory Module 1 Alice World ▶ Module 1 Discussion Questions ▶ Lab 1 Alice World Projects (3)

Guiding Questions

▶ **What are computing and computational thinking and why are they important to learn?** ▶ **What is computer programming?** ▶ **Why is creating a plan important for creating a program?** ▶ **Why do experienced programmers use prediction when creating and writing programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 10-11) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.3.1, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.2.1, 5.4.1.**	**Web** Guides Through Alice 2, "1. Telling a Story: Lab" ("Mod1.2LabStarter.a2W" for Lab 1B and "Mod1.3LabStarter.a2W" for Lab 1C, Lab Part A, Lab Part B) **or** Computer Science Principles Fall 2015, "Lab 1A," "Lab 1B," and "Lab 1C" **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 1A** Students create an Alice world. They learn to select a virtual world, insert, move, and set up objects (characters) via the object pane, and use the built-in methods provided to move the objects around in the world. Students answer two questions about the objects and methods used. Students use a template and create an MLA-formatted document titled "Lab 1 Questions" to answer the questions for Lab 1. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-D; **1.3.1** E; **5.1.1** A; **5.1.2** A, B, D

Students are expected to demonstrate mastery of the concepts in the modules by completing labs and projects. Students are expected to come prepared to lab, having done the relevant exploratory projects and assignments and having engaged with and learned the material in lecture and online modules.

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:
▶ Exploratory Module 1 Alice World ▶ Module 1 Discussion Questions ▶ Lab 1 Alice World Projects (3)

Guiding Questions

▶ **What are computing and computational thinking and why are they important to learn?** ▶ **What is computer programming?** ▶ **Why is creating a plan important for creating a program?** ▶ **Why do experienced programmers use prediction when creating and writing programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 10-11) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.3.1, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.2.1, 5.4.1.**		**Instructional Activity: Lab 1B** Students use the Alice world from Lab 1A to complete Lab 1B by writing code based upon the scenario/story described for Lab 1B. Students explain the **Do in order** block/tile (control structure) and why actions happen in the order they do (one before another). Students also manipulate action in the program by moving method tiles (instructions) based on viewing an Alice program (code). Students explain how the **Do together** block/tile (control structure) can be used to make actions happen at the same time. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **4.1.1** A, B; **4.1.2** B, C; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.3** A, B, D, F; **5.1.2** A-C, H-J; **5.2.1** B-D; **5.4.1** A, E, F, H, I, K, L
		Instructional Activity: Lab 1C Students write code based upon the scenario/story described for Lab 1C. They explain why actions happen in the order they do (one before another) and how to manipulate action by moving method tiles (instructions) based on looking at an Alice program. Students also explain how the **Do together** tile (control structure) can be used to make actions happen at the same time. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.3.1** A-D; **4.1.1** A, B; **4.1.2** B, C; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.3** A, B, D, F; **5.1.2** A-C, H-J; **5.2.1** B-D; **5.4.1** A, E, F, H, I, K, L

◀ *A key instructional component is using pair programming. Another key component is having students explain their code to one another and then also explain it to me (clearly!). Again, emphasize to students the importance of reading the instructions carefully and not skimming the material.*

Lab 1A, Lab 1B, and Lab 1C provide step-by-step instructions for this activity.

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:

- Exploratory Module 1 Alice World
- Module 1 Discussion Questions
- Lab 1 Alice World Projects (3)

Guiding Questions

- What are computing and computational thinking and why are they important to learn?
- What is computer programming?
- Why is creating a plan important for creating a program?
- Why do experienced programmers use prediction when creating and writing programs?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 10-11) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.3.1, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.2.1, 5.4.1.**	**Web** Guides through Alice 2, "1. Telling a Story: Practice Problems" ("Beetles dance v1.0") Computer Science Principles Fall 2015, "Beetle Dance Module 1 Extra Programming Problem"	**Instructional Activity: Practice Programming Problems** Students create an Alice world that requires them to set up the world, add five objects to it, and use built-in methods: **turn**, **roll**, **say**, **Do in order**, and **Do together**. Students watch a video and emulate what it shows with the "BeetlesDance.a2w" Alice world they are creating. As an extra challenge, students view a second video — "Beetles Dance with Fish" — and emulate it by adding more objects and using "duration" for timing purposes. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A- F; **1.2.5** A-D; **1.3.1** D; **4.1.1** A, B; **4.1.2** B, C; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.3** A, B; D, F; **5.1.2** A-C, H-J; **5.2.1** B-D; **5.4.1** A, E, F, H, K, L
	Web CANVAS LMS, Google Sites, or any platform for creating and hosting websites Expeditions through Alice, "1. Telling a Story"	**Formative Assessment: Digital Portfolio — Reflection (Essential Questions)** Students create a digital portfolio (website) on an appropriate platform (e.g., Google Sites, Canvas). They upload their work and write reflections and/or explanations of what they have learned, including the process of learning when creating the Alice worlds. Students then write a reflection based on the "Things you will learn in this module" on the first page of module 1 in Expeditions Through Alice. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A- F; **1.2.5** A-D; **1.3.1** A; **4.1.1** A, B; **4.1.2** B, C; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-C, H-J; **5.2.1** B, D, E; **5.4.1** A, E, F, H, K, L

In this assessment, students must demonstrate understanding and mastery of the Module 1 Alice World. I provide feedback and students can improve their portfolios based on my comments.

BIG IDEA 1 Creativity
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.1, EU 1.2, EU 1.3, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.4, EU 7.1, EU 7.3, EU 7.4

Projects and Major Assignments:

▶ Exploratory Module 1 Alice World ▶ Module 1 Discussion Questions ▶ Lab 1 Alice World Projects (3)

Guiding Questions

▶ **What are computing and computational thinking and why are they important to learn?** ▶ **What is computer programming?** ▶ **Why is creating a plan important for creating a program?** ▶ **Why do experienced programmers use prediction when creating and writing programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.3.1:** Analyze the beneficial and harmful effects of computing. [P4] **LO 7.4.1:** Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts. [P1]	**Web** Blogging tools (Blogger, Wordpress, Tumblr) "Growing Up Online"	**Instructional Activity: The Internet as a Social Experience** Students participate in a discussion of online security issues and student use of social networking sites. I show chapters 1-3 of the video "Growing Up Online" from the PBS series Frontline. Then students participate in another discussion, focusing on the content of the video, which includes social networking sites, blogs, email, online chatting, and the kind of impact these applications have had on students' lives. Students set up a blog and create two blog entries — one describing some of their current online experiences and the other on the PBS video. Students should mention if any of their thoughts about online security issues or social networking applications changed based on the video or discussions. **Essential knowledge addressed: 7.1.1** A-C, H, M; **7.3.1** B, D, G, I-K; **7.4.1** A-E

This activity is based on Big Idea 7: Global Impact.

UNIT 2: TELLING AND DIVIDING A STORY (METHODS)

Estimated Time: 15–18 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

▸ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

▸ Exploratory modules 2 and 3 Alice Worlds ▸ Labs 2 and 3 Alice World Projects ▸ Module 3 Extra Programming Problem

Guiding Questions

▸ **What are the advantages of using methods within computer programs?** ▸ **Why is abstraction a key concept for programmers to understand and use in computer programs?** ▸ **How do we use abstraction in our daily lives?** ▸ **How are methods applied as a concept of abstraction in computing and computer programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2] **LO 2.2.2:** Use multiple levels of abstraction to write programs. [P3]	**Print** Dann, Cooper, and Pausch, p. 27 – 41	**Instructional Activity: Creating an Alice World with Methods** Students create an Alice world that requires the use of methods. I demonstrate or model the process of creating methods for an Alice world with prewritten code and students download the same world. I explain abstraction and how it relates to the concept of methods. Students think of a word such as "chair" and describe what pops into their mind. Usually it's a mental image of a chair. We discuss the complexity of describing a chair, such as "seat," "legs," etc.; we discuss the similarity of the "chair" example to a method and the "method name" that uses abstraction to "hide" away the complexity of instructions. Students work in pairs and create two new methods with appropriate method names based on what they learned from the demonstration. **Essential knowledge addressed:** **1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B; **2.2.3** B; **2.3.1** A-C; **4.1.1** A, B; **4.1.2** A-C; I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-C; **5.4.1** A-F, H- L

UNIT 2: TELLING AND DIVIDING A STORY (METHODS)

Estimated Time: 15–18 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:
- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:
- Exploratory modules 2 and 3 Alice Worlds
- Labs 2 and 3 Alice World Projects
- Module 3 Extra Programming Problem

Guiding Questions

- What are the advantages of using methods within computer programs?
- Why is abstraction a key concept for programmers to understand and use in computer programs?
- How do we use abstraction in our daily lives?
- How are methods applied as a concept of abstraction in computing and computer programs?

Learning Objectives

LO 2.2.3: Identify multiple levels of abstractions that are used when writing programs. [P3]

LO 2.3.1: Use models and simulations to represent phenomena. [P3]

LO 4.1.1: Develop an algorithm for implementation in a program. [P2]

LO 4.1.2: Express an algorithm in a language. [P5]

LO 4.2.4: Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]

LO 5.1.1: Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2]

LO 5.1.2: Develop a correct program to solve problems. [P2]

LO 5.1.3: Collaborate to develop a program. [P6]

LO 5.2.1: Explain how programs implement algorithms. [P3]

LO 5.3.1: Use abstraction to manage complexity in programs. [P3]

LO 5.4.1: Evaluate the correctness of a program. [P4]

Materials / Instructional Activities and Classroom Assessments

Materials

Web
Expeditions through Alice, "2. Dividing the Story (methods)" ("FirstWorldV1.a2w" or "Mod2.1Starter.a2w")

Guides to Alice 2, "2. Dividing the Story (methods)" (video explanations for sections 2.3–2.5)

Instructional Activity: Exploratory Module 2 Alice World

Students work in pairs to create and complete the exploratory Module 2 Alice world. The module 2 reading directs students to **download** and create an Alice world, typing a specific set of instructions or code that demonstrates and uses the advantages of methods. The instructions also demonstrate the concept of abstraction by creating methods. The original program uses several lines of code; thus, by creating and using methods (or procedures), the program decreases the several lines of code to only three lines of code. The three lines of codes are the methods that have been created.

Essential knowledge addressed: **1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B; **2.2.3** B; **4.1.1** A, B; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D; **5.4.1** A-F, H-L

Students learn that methods break up a large amount of code into smaller, more manageable pieces; by default, Alice executes the method "world.my first method" when you press "play." Methods must be called in "world.my first method" to be executed. Breaking up programs is one way humans use abstraction to manage complexity in computing: we need to use abstraction to manage thinking about all the instructions given in a computer program.

Materials

Web
Computer Science Principles Fall 2015, "Module 2 Questions Assignment"

Software
Google Docs or Microsoft Word

Instructional Activity: Exploratory Module 2 Questions

As students are reading and following the instructions for creating the exploratory Module 2 Alice world, they answer a series of questions in the module that promote a deeper understanding of the computing concepts (methods) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions are provided.

Essential knowledge addressed: **1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B, **2.2.3** B; **4.1.1** A, B; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.2.1** A, B; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D; **5.4.1** A-F, H-L

Students are reminded not to skim the reading and to think analytically about their answers. I demonstrate to students what a good versus bad answer looks like.

UNIT 2: TELLING AND DIVIDING A STORY (METHODS)

Estimated Time: 15–18 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

- Exploratory modules 2 and 3 Alice Worlds
- Labs 2 and 3 Alice World Projects
- Module 3 Extra Programming Problem

Guiding Questions

▸ What are the advantages of using methods within computer programs? ▸ Why is abstraction a key concept for programmers to understand and use in computer programs? ▸ How do we use abstraction in our daily lives? ▸ How are methods applied as a concept of abstraction in computing and computer programs?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 19-20) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1.**	**Software** Classroom-Response System **Web** Guides Through Alice 2, "2. Dividing the Story (methods): Lecture" ("Module 2 Dividing The Story.pptx")	**Summative Assessment: Module 2 Quiz** Using a classroom-response system, students take an assessment composed of three multiple-choice questions. The questions are based on methods (procedures) and why methods are used within programming. **Essential knowledge addressed: 2.2.1** A, B; **5.3.1** B-D; **5.4.1** C-E
		Formative Assessment: Module 2 Peer-Discussion Questions Students work in groups of three to answer three multiple-choice questions using a classroom-response system. Students read the question and then select an answer. They then discuss their answers with their groups, explaining clearly why they chose their answer and more importantly why they did not choose the other answers. Students volunteer to share their responses with the class. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B, **2.2.3** B; **4.1.1** A, B; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.2.1** A, B; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D; **5.4.1** A,C-F, H-L

This summative assessment addresses the following guiding questions:

- What are the advantages of using methods within computer programs?
- Why is abstraction a key concept for programmers to understand and use in computer programs?

Feedback is provided for every question; students and I discuss which answer is correct and which ones are incorrect. Students further develop technical analysis and communication skills with this pedagogical technique.

UNIT 2: TELLING AND DIVIDING A STORY (METHODS)

Estimated Time: 15–18 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:
▶ Exploratory modules 2 and 3 Alice Worlds ▶ Labs 2 and 3 Alice World Projects ▶ Module 3 Extra Programming Problem

Guiding Questions

▶ **What are the advantages of using methods within computer programs?** ▶ **Why is abstraction a key concept for programmers to understand and use in computer programs?** ▶ **How do we use abstraction in our daily lives?** ▶ **How are methods applied as a concept of abstraction in computing and computer programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 19-20) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1.**	**Web** Guides Through Alice 2, "2. Dividing the Story (methods): Lab" (instructions and "Mod2 LabStarter.a2W") Computer Science Principles Fall 2015, "Lab: Module 2 Dividing the Story (Methods)" **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 2** In Lab 2, students experience reading step-by-step instructions and playing around with code written by someone else, with the goal of understanding what the code does and being able to modify it. Students work in pairs and are able to create a new method in Alice. They divide a provided program into logical methods to manage complexity through abstraction. They also answer a series of lab questions about methods and abstraction based on the Alice world they modify. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B, **2.2.3** B; **2.3.1** A-D; **4.1.1** A, B; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D; **5.4.1** A,C-F, H-L
	Web Guides Through Alice 2, "2. Dividing the Story (methods): Practice Problems" ("Mod2_PP1.a2w," "Animal Dance (v1.0)," and videos of "Animal Dance" Alice Worlds) Computer Science Principles Fall 2015, "Module 2 Practice Problem: Animal Dance"	**Instructional Activity: Module 2 Practice Programming Problems** Students watch a video of animals dancing in an Alice world then modify the world through separating the code for one large dance into smaller dance pieces by creating methods. The first Alice world is named AnimalDanceV1.a2w. Then students create and add their own dance sequence as a method. They name this world: AnimalDanceV2.a2w. Students also create an Alice world of their own choosing that requires them to break up code into methods with the appropriate method names. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B, **2.2.3** B; **4.1.1** A, B; F,G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D

Students run their code and explain to me what happens: they match the code with the executed behavior. I read their code to ensure their explanations are correct. I check the methods, and the students explain how abstraction was used. Answers are also required in written form.

UNIT 2: TELLING AND DIVIDING A STORY (METHODS)

Estimated Time: 15–18 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:
▶ Exploratory modules 2 and 3 Alice Worlds ▶ Labs 2 and 3 Alice World Projects ▶ Module 3 Extra Programming Problem

Guiding Questions

▶ What are the advantages of using methods within computer programs? ▶ Why is abstraction a key concept for programmers to understand and use in computer programs? ▶ How do we use abstraction in our daily lives? ▶ How are methods applied as a concept of abstraction in computing and computer programs?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2] **LO 2.2.2:** Use multiple levels of abstraction to write programs. [P3]	**Web** Expeditions Through Alice, 3. Stories from Pieces (methods)" Guides Through Alice 2, "3. Stories from Pieces: Book" (explanation of code for sections 3.2–3.4 and "FirstWorldV2.a2w" or "Mod3.1Starter.a2w")	**Instructional Activity: Exploratory Module 3 Alice World** Students work in pairs to create and complete the exploratory Module 3 Alice world. The Module 3 reading directs students to create an Alice world and type a specific set of instructions or code that demonstrates and uses the advantages of methods. Students reuse, rearrange, and duplicate methods to create different "stories" or Alice worlds. Students add comments to document and describe the purpose of the code for this and all future Alice worlds. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B, **2.2.3** B; **2.3.1** A-C; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D; **5.4.1** A-F, H-L
	Web Computer Science Principles Fall 2015, Module 3 Questions Assignment **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 3 Questions** As students are reading and following the instructions for creating the exploratory Module 3 Alice world, they answer a series of questions in the module that promote a deeper understanding of the computing concepts (methods) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions are provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B; **2.2.3** B; **4.1.1** A, B,E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.2.1** A-E; **5.3.1** A-D

This module demonstrates to students the concept of abstraction and the advantages of using methods (they can be reused and rearranged in different orders to produce different results). Students are introduced to the importance of commenting. As students are working, I circulate asking questions about the code they are modifying; they must clearly explain to their partner and me what they are doing and learning, including the process of creating methods and abstraction.

UNIT 2: TELLING AND DIVIDING A STORY (METHODS)

Estimated Time: 15–18 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

- Exploratory modules 2 and 3 Alice Worlds
- Labs 2 and 3 Alice World Projects
- Module 3 Extra Programming Problem

Guiding Questions

- **What are the advantages of using methods within computer programs?**
- **Why is abstraction a key concept for programmers to understand and use in computer programs?**
- **How do we use abstraction in our daily lives?**
- **How are methods applied as a concept of abstraction in computing and computer programs?**

Learning Objectives

LO 2.2.3: Identify multiple levels of abstractions that are used when writing programs. [P3]

LO 4.1.1: Develop an algorithm for implementation in a program. [P2]

LO 4.1.2: Express an algorithm in a language. [P5]

LO 4.2.4: Evaluate algorithms analytically and empirically for efficiency, correctness, and clarity. [P4]

LO 5.1.1: Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2]

LO 5.1.2: Develop a correct program to solve problems. [P2]

LO 5.1.3: Collaborate to develop a program. [P6]

LO 5.2.1: Explain how programs implement algorithms. [P3]

LO 5.3.1: Use abstraction to manage complexity. [P3]

LO 5.4.1: Evaluate the correctness of a program. [P4]

Materials	Instructional Activities and Classroom Assessments	
Software Classroom-Response System **Web** Guides Through Alice 2, "3. Stories from Pieces: Lecture" ("Module 3 Stories From Pieces.pptx")	**Summative Assessment: Module 3 Quiz** Using a classroom-response system, students take an assessment composed of just one multiple-choice question. **Essential knowledge addressed: 2.2.1** A, B; **5.4.1** C-E	*This module was just a small extension of Module 2, emphasizing that once you have methods you can do a lot of work and make flexible stories by calling a method more than once or by reordering how you call methods.* *This summative assessment addresses the following guiding question:* ▸ What are the advantages of using methods within computer programs?
Software Classroom-Response System **Web** Guides Through Alice 2, "3. Stories from Pieces: Lecture" ("Module 3 Stories From Pieces.pptx")	**Formative Assessment: Module 3 Peer Discussion Questions** Students work in groups of three to answer one multiple-choice question using a classroom-response system. Students read the question, select an answer, and then discuss their answers with their groups; they must clearly explain why they chose their answer and more importantly, why they did not choose the other answers. Students volunteer to share their responses with the class. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B, **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D	*I share the correct answer and explain clearly why that answer is correct and why the other ones are not.* *As with all of the peer-discussion questions, students are developing technical, analytical, and communication skills with this pedagogical technique.*

UNIT 2: TELLING AND DIVIDING A STORY (METHODS)

Estimated Time: 15–18 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

▶ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

▶ Exploratory modules 2 and 3 Alice Worlds ▶ Labs 2 and 3 Alice World Projects ▶ Module 3 Extra Programming Problem

Guiding Questions

▶ **What are the advantages of using methods within computer programs?** ▶ **Why is abstraction a key concept for programmers to understand and use in computer programs?** ▶ **How do we use abstraction in our daily lives?** ▶ **How are methods applied as a concept of abstraction in computing and computer programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the activities on pages 23-24 are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.2.2, 2.2.3, 4.1.1, 4.1.2, 4.2.4, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1.**	**Web** Guides Through Alice 2, "3. Stories from Pieces: Lab" (Step-by-step instructions and "Lab2.a2W" or "Mod3LabStarter.a2w") Computer Science Principles Fall 2015, "Lab 3 Instructions" **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 3** For Lab 3, students work in pairs to modify the code for the Lab 2 Alice world. Students learn to reuse methods to create variations on a "story" (program) by calling methods in different orders and/or calling a method more than once. Students experience how reusing a method (calling it more than once) can lead to the need to redesign or change code in small ways. Students then answer a series of lab questions about methods and abstraction based on the Alice world they modify. Comments are required for this Alice world. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B, **2.2.3** B; **4.1.1** A, B; **4.1.2** A-C; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D
	Web Guides to Alice 2, "3. Stories from Pieces: Practice Problems" ("Ice Skater Routine (v1.0)," "Mod3PP1_Starter.a2w," and videos of "Ice Skater Routine") Computer Science Principles Fall 2015, "Module 3 Practice Problem: Ice Skater Routine"	**Instructional Activity: Module 3 Practice Programming Problems** Students download the "Mod3PP1_Starter.a2w" Alice world and watch a video about an ice skater performing ice skating routines. The methods for these routines have already been created in the downloaded Alice world. Students make calls to the ice skater methods to mimic the Alice world that has been shown in the video. For a further challenge, another video is provided with extra routines. Students then make more calls to mimic the Alice world shown in the second video. Comments are required for this Alice world. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A, B; **2.2.2** B, **2.2.3** B; **2.3.1** A-D; **4.1.1** A, B; **4.1.2** A-C; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** B, C, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-D; **5.4.1** A-F, H-L

UNIT 3: BEHAVING THE SAME VS. DIFFERENTLY (PARAMETERS)

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Create Exploratory Modules 4 and 5 Alice Worlds
- Labs 4 and 5 Alice World Projects

Guiding Questions

- What are parameters and how are they used in computer programs?
- What are the advantages of using parameters within computer programs?
- How are parameters applied as a concept of abstraction in computing and computer programs?
- How are parameters used in our daily lives? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2]	**Web** Expeditions Through Alice, "4. Acting the Same (parameters)" ("Mod4.2 Starter.a2w") Guides Through Alice 2, "4. Acting the Same (parameters): Book" (explanation of code for sections 4.4–4.7)	**Instructional Activity: Exploratory Module 4 Alice World** Students work in pairs to create and complete the exploratory Module 4 Alice world. Students are directed to modify an Alice world and type a specific set of instructions or code that demonstrates and uses the advantages of parameters (values that are added to a method call). Students create parameters for methods and learn that a method can behave differently by using different types of parameters. Students learn how to specify that a method will expect to have parameter values supplied when the method is called and how to call a method that expects parameter values. Students also learn how the concept of abstraction is applied to parameters. Comments are required for this program. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-C; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A
	Web Computer Science Principles Fall 2015, "Module 4 Questions Assignment" **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 4 Questions:** As students are reading and following the instructions for creating the exploratory module 4 Alice world program, they answer a series of questions in the module that promote a deeper understanding of the computing concepts (parameters) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions for this activity are provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A,B, D; **5.1.2** A-F, H-J; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A

UNIT 3: BEHAVING THE SAME VS. DIFFERENTLY (PARAMETERS)

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Create Exploratory Modules 4 and 5 Alice Worlds
- Labs 4 and 5 Alice World Projects

Guiding Questions

- **What are parameters and how are they used in computer programs?**
- **What are the advantages of using parameters within computer programs?**
- **How are parameters applied as a concept of abstraction in computing and computer programs?**
- **How are parameters used in our daily lives? What are some examples?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 2.3.1: Use models and simulations to represent phenomena. [P3] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Software** Classroom-Response System **Web** Guides Through Alice 2, "4. Acting the Same (parameters): Lecture" ("Module 4 Acting The Same. pptx")	**Summative Assessment: Module 4 Quiz** Using a classroom-response system, students take an assessment composed of two multiple-choice questions. **Essential knowledge addressed: 5.3.1** A-G; **5.4.1** C-F, H-L; **5.5.1** A	*This summative assessment addresses the following guiding questions:* - What are parameters and how they are used in computer programs? - What are the advantages of using parameters within computer programs?
	Software Classroom-Response System **Web** Guides Through Alice 2, "4. Acting the Same (parameters): Lecture" (all files on page)	**Formative Assessment: Module 4 Peer Discussion Questions** Students work in groups of three to answer several multiple-choice questions using a classroom-response system. Students read the question, select an answer, and then discuss their answers with their groups. They should clearly explain why they chose their answer and more importantly why they did not choose the other answers. Students are then asked to volunteer and share their responses with the class. I then share with them the correct answer and explain clearly why that answer is correct and the other ones are not. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A	*For each question, the classroom-response system is used to display histograms with recorded answers. The first histogram displays a chart with how students answer individually and a second histogram displays their answers after the discussion with their group. The answers are compared and recorded within the system. Students receive immediate feedback and clear explanations for both the correct and incorrect answers.*

UNIT 3: BEHAVING THE SAME VS. DIFFERENTLY (PARAMETERS)

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Create Exploratory Modules 4 and 5 Alice Worlds
- Labs 4 and 5 Alice World Projects

Guiding Questions

- What are parameters and how are they used in computer programs?
- What are the advantages of using parameters within computer programs?
- How are parameters applied as a concept of abstraction in computing and computer programs?
- How are parameters used in our daily lives? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 26-27) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.3.1, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Web** Guides Through Alice 2, "4. Acting the Same (parameters): Lab" (Lab 4 as a Googledoc and "Lab3.a2W" or "Mod4LabStarter .a2w") Computer Science Principles Fall 2015, Lab 4 **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 4** For Lab 4, students work in pairs to modify the code for the Lab 3 Alice world. Based on a scenario description, students create a new method (without guidance or suggestions) to modify the story (program). Students modify a method to take a parameter to control which object does the acting. Students then answer a series of lab questions about parameters based on the Alice world they modify. Comments are required. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-D; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A,B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A
	Web Guides Through Alice 2, "4. Acting the Same (parameters): Practice Problems" ("Fish Lives (v1.0)," "Mod4PP1_Starter .a2w," Videos of "Ice Skater Routine") **Software** Computer Science Principles Fall 2015, "Module 4 Practice Problem: Fish Lives"	**Instructional Activity: Module 4 Practice Programming Problems** Students download the "Mod4PP1_Starter.a2w" Alice world and view a video about the "Fish Lives" Alice world. A screenshot is included in the instructions showing the methods for the "Fish Lives" Alice world. Students create the methods, mimicking the actions that occur in the video, and create parameters to select the fish objects. As an extra challenge, students create another parameterized method that has the fish dance before a shark attacks. Comments are required. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A

UNIT 3: BEHAVING THE SAME VS. DIFFERENTLY (PARAMETERS)

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

- Create Exploratory Modules 4 and 5 Alice Worlds
- Labs 4 and 5 Alice World Projects

Guiding Questions

- What are parameters and how are they used in computer programs?
- What are the advantages of using parameters within computer programs?
- How are parameters applied as a concept of abstraction in computing and computer programs?
- How are parameters used in our daily lives? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2]	**Web** Expeditions Through Alice, "5. Acting Differently (parameters)" Guides Through Alice 2, "5. Acting Differently (parameters): Book" (explanation of code for sections 5.2–5.4)	**Instructional Activity: Exploratory Module 5 Alice World** Students work in pairs to create and complete the exploratory Module 5 Alice world. They modify an Alice world and type a specific set of instructions or code. Students create and add more than one parameter for a method. They also create different types of parameters, including numerical or object-based ones. Comments are required. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-C; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A,B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A
	Web Computer Science Principles Fall 2015, "Module 5 Questions Assignment" **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 5 Questions** As students are reading and following the instructions for creating the exploratory Module 5 Alice world program, they are also required to answer a series of questions provided in the online book for Module 5 that promotes a deeper understanding of the computing concepts (*parameters*) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions for this activity are provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** A, B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A

Students learn that a method can take any number of parameters. They learn that parameters can have different types of values, such as numbers or objects. This can be useful in specifying variation for methods. Students also learn how the concept of abstraction is applied to parameters.

UNIT 3: BEHAVING THE SAME VS. DIFFERENTLY (PARAMETERS)

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▸ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

▸ Create Exploratory Modules 4 and 5 Alice Worlds ▸ Labs 4 and 5 Alice World Projects

Guiding Questions

▸ What are parameters and how are they used in computer programs? ▸ What are the advantages of using parameters within computer programs? ▸ How are parameters applied as a concept of abstraction in computing and computer programs? ▸ How are parameters used in our daily lives? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 4.1.1: Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Software** Classroom-Response System **Web** Guides Through Alice 2, "5. Acting Differently (parameters): Lecture" (Module 5 Acting Differently.pptx)	**Summative Assessment: Module 5 Quiz** Using a classroom-response system, students take an assessment composed of three multiple-choice questions. **Essential knowledge addressed: 5.3.1** A-G; **5.4.1** C-F, H-L; **5.5.1** A	*This summative assessment addresses the following guiding questions:* ▸ What are parameters and how they are used in computer programs? ▸ What are the advantages of using parameters within computer programs?
	Software Classroom-Response System **Web** Guides Through Alice 2, "5. Acting Differently (parameters): Lecture" (all files on this page)	**Formative Assessment: Module 5 Peer-Discussion Questions** Students work in groups of three to answer several multiple-choice questions using a classroom-response system. Students read the question, select an answer, and discuss their answer with their groups; they must explain why they chose their answer, and, more importantly, explain why they did not choose the other answers. Students then volunteer to share their responses with the class. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A	*I share the correct answer with the students, and we discuss why that answer is correct and why the other ones are not.*

UNIT 3: BEHAVING THE SAME VS. DIFFERENTLY (PARAMETERS)

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

▶ Create Exploratory Modules 4 and 5 Alice Worlds ▶ Labs 4 and 5 Alice World Projects

Guiding Questions

▶ What are parameters and how are they used in computer programs? ▶ What are the advantages of using parameters within computer programs? ▶ How are parameters applied as a concept of abstraction in computing and computer programs? ▶ How are parameters used in our daily lives? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the activities on pages 29-30 are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.3.1, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Web** Guides Through Alice 2, "5. Acting Differently (parameters): Lab" ("Lab as Googledoc," "Mod5Lab Starter.a2w") Computer Science Principles Fall 2015, "Lab 5 Instructions"	**Instructional Activity: Lab 5** For Lab 5, students work in pairs to modify the code for the Lab 4 Alice world. Students modify a method to take more than one parameter including parameters of type **Number** and **Object**. Students add three additional parameters to a method: two of the parameters are given and students create one of their own. Comments are required for this world. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-D; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A,B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A
	Web Guides Through Alice 2, "5. Acting Differently (parameters): Practice Problems" (Ice Skater 2 (v1.0) and Starter World) Computer Science Principles Fall 2015, "Mod 5 Practice Programming Problem: Ice Skater 2 v1.0"	**Instructional Activity: Module 5 Practice Programming Problems** Students either download the "Mod5PP1_Starter.a2w" Alice world or use the previous world they created: IceSkaterV1.0.a2w. They watch two different videos and have to figure out what is different between them. Students are asked to notice that the routine the ice skater is performing does not change (the order of the tricks), but the details of the tricks have changed. Students create and add parameters to the methods so that they can mimic what is happening in the second video. After modifying the code, they view a third video. They make the routine shown in video three within the Alice world by only changing the values of the parameters in "world.my first method." Comments are required for this program. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A

◀ *Pair programming is a key component for students to understand and complete this lab. Students run their code and explain to me what happens — they match the code with the executed behavior. I read their code to ensure their explanations are correct, check the methods and parameters, and students include an explanation of how abstraction was used.*

UNIT 3: BEHAVING THE SAME VS. DIFFERENTLY (PARAMETERS)

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming
BIG IDEA 7 Global Impact

Essential Understandings:

▶ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5, EU 7.1, EU 7.2, EU 7.3, EU 7.5

Projects and Major Assignments:

▶ Create Exploratory Modules 4 and 5 Alice Worlds ▶ Labs 4 and 5 Alice World Projects

Guiding Questions

▶ What are parameters and how are they used in computer programs? ▶ What are the advantages of using parameters within computer programs? ▶ How are parameters applied as a concept of abstraction in computing and computer programs? ▶ How are parameters used in our daily lives? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
	Web Guides Through Alice 2, "5. Acting Differently (parameters): Practice Problems" (Ice Skater 2 (v1.0)	**Instructional Activity: Module 5 Open-Ended or Group Type Practice Programming Problems** Students create a world of their choosing that involves a lot of little parts broken up into methods; the methods can be affected by lots of different small changes, and a parameter is used to determine the specifics. For example, students could build a world where a ball bounces 5 feet on the first bounce, 4 feet on the second bounce, 3 on the third, 2 on the fourth, 1 foot on the fifth, then stops. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A, B; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** A, B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-F, H-L; **5.5.1** A	*Students are challenged to create programs using the computing concepts they have learned in this course. They think computationally as they create their own animated stories.*
LO 7.1.1: Explain how computing innovations affect communication, interaction, and cognition. [P4] **LO 7.1.2:** Explain how people participate in a problem-solving process that scales. [P4] **LO 7.2.1:** Explain how computing has impacted innovations in other fields. [P1] **LO 7.3.1:** Analyze the beneficial and harmful effects of computing. [P4] **LO 7.5.2:** Evaluate online and print sources for appropriateness and credibility. [P5]	**Web** Guides Through Alice 2, "Technology and Society Assignments" (Quality of information on web, analysis of risks of technology - TED talks, USCD rubric, UCSD example) TED.com	**Instructional Activity: "Technology: It's Not Inherently Good or Bad"** In this activity, students analyze technological issues in the world around them. They select a Ted Talk on a technological innovation that interests them and then write a reflection on it, including its potential benefits and harmful effects on people and society. Students submit their reflection on a discussion board, and then read 10 other submissions and respond to two of their peers' submissions. **Essential knowledge addressed: 7.1.1** E, M, N; **7.1.2** A-G; **7.2.1** A-C; **7.3.1** A, E, G, K; **7.5.2** A, B	*Many students have not participated on a discussion board and need guidance: I explain that their reflections and responses must be analytical, articulate, and clear. I give them examples of the dos and don'ts; excellent videos on the use of discussion boards can be found on YouTube. This is practice for the upcoming Explore Performance Task.*

UNIT 4: GETTING INTO THE STORY (EVENTS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

- Create Exploratory Module 6 Alice Worlds
- Lab 6 Alice World Projects
- Module 6 Extra Programming Problem

Guiding Questions

- **What are events and how are they used in interactive computer programs?**
- **How would you describe the different kinds of events that enable a user to interact with computer programs in different ways?**
- **What is an event handler and how does the event handler allow events to call methods?**
- **How can you write code so that it is reused to handle more than one event?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2]	**Web** Expeditions Through Alice, "6: Get Into the Story (events)" Guides Through Alice 2, "6. Get Into the Story (events): Book" (explanation of code sections 6.4–6.6)	**Instructional Activity: Exploratory Module 6 Alice World** In this activity students learn what events are and how they can work with them to enable their audience to interact with programs. Students work in pairs to create and complete the exploratory Module 6 Alice world. They modify three Alice worlds and type a specific set of instructions or code. Students make the Alice worlds interactive by adding events, event handlers, methods, and parameters. Students learn that there are a number of different kinds of events, such as using mouse or keyboard buttons, that enable users to interact with their programs in different ways. Students also learn how the code they write to respond to an event can, in some situations, be reused to handle many other events. Comments are required for this program. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-D; **4.1.1** A, B, E-G; **4.1.2** A-C; I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-N; **5.5.1** A
	Web Computer Science Principles Fall 2015, "Module 6 Questions Assignment Events" **Software** Google Docs Microsoft Word	**Instructional Activity: Exploratory Module 6 Questions** As students are reading and following the instructions for creating the exploratory Module 6 Alice world program, they answer a series of questions in the module that promote a deeper understanding of the computing concepts (events) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions for this activity are provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C; I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-N; **5.5.1** A

This activity helps students gain a deeper understanding of events, event handlers, key press events, and mouse press events. They learn how to make computer programs interactive by understanding that an event handler is the connection between a particular event and the code (a method) called by that event.

UNIT 4: GETTING INTO THE STORY (EVENTS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

▸ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

▸ Create Exploratory Module 6 Alice Worlds ▸ Lab 6 Alice World Projects ▸ Module 6 Extra Programming Problem

Guiding Questions

▸ What are events and how are they used in interactive computer programs? ▸ How would you describe the different kinds of events that enable a user to interact with computer programs in different ways? ▸ What is an event handler and how does the event handler allow events to call methods? ▸ How can you write code so that it is reused to handle more than one event?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 2.3.1: Use models and simulations to represent phenomena. [P3] **LO 2.3.2:** Use models and simulations to formulate, refine, and test hypotheses. [P3] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4]	**Software** Classroom-Response System **Web** Guides Through Alice 2, "6. Get Into the Story (events): Lecture" ("Module 6 Get Into The Story.pptx")	**Summative Assessment: Module 6 Quiz with Classroom-Response System** Using a classroom-response system, students take an assessment composed of three multiple-choice questions. **Essential knowledge addressed: 5.3.1** A-G; **5.4.1** C-N; **5.5.1** A	*This summative assessment addresses all of the guiding questions for the unit.*
	Software Classroom-Response System **Web** Guides Through Alice 2, "6. Get Into the Story (events): Lecture" (all files on page)	**Formative Assessment: Module 6 Peer Discussion Questions** Students work in groups of three to answer a few multiple-choice questions using a classroom-response system. Students read the question, select an answer, and then discuss their answers with their groups; they must clearly explain why they chose their answer and, more importantly, why they did not choose the other answers. **Essential knowledge addressed: EK 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F; H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-N, **5.5.1** A	*Students volunteer their responses with the class. I share with them the correct answer and give them feedback on why that answer is correct and the other ones are not.*

UNIT 4: GETTING INTO THE STORY (EVENTS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

- Create Exploratory Module 6 Alice Worlds
- Lab 6 Alice World Projects
- Module 6 Extra Programming Problem

Guiding Questions

- What are events and how are they used in interactive computer programs?
- How would you describe the different kinds of events that enable a user to interact with computer programs in different ways?
- What is an event handler and how does the event handler allow events to call methods?
- How can you write code so that it is reused to handle more than one event?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the activities on pages 33-34 are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.3.1, 2.3.2, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1.**	**Web** Guides Through Alice 2, "6. Get Into the Story (events): Lab" (Lab 6 as Googledoc, "Mod6:LabStarter. a2w") Computer Science Principles Fall 2015, "Lab 6 Instructions" **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 6** Students learn to create and set up events (specifically "when a key is pressed" events) to control the actions of an object. Working in pairs, students set up a simple flight simulator with five events to control the plane; the events are "up," "down," "left arrow" to go left, "right arrow" to go right, and "spacebar" to go forward. Students then answer a series of lab questions about events based upon the Alice program they created. Students run their code and explain to me what happens. I read their code to ensure their explanations are correct. I also check the events, event handlers, and methods; students include an explanation of how abstraction was used. Comments are required for this program. **Essential knowledge addressed: EK 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-D; **2.3.2** A-H; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B; D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-N; **5.5.1** A
	Web Guides Through Alice 2, "6. Get Into the Story (events): Practice Problems" (all files on page) Computer Science Principles Fall 2015, "Mod 6 Practice Programming Exercise: Wind Up Bunny"	**Instructional Activity: Module 6 Practice Programming Problem** Students download the "Mod6PP1_Starter.a2w" Alice world and view the video. They then mimic the actions of the objects by creating and using a mouse click event and the methods that the event handlers call for in the "Wind-Up Bunny" program. Students first create two methods after viewing the actions that the bunny performs when the mouse is clicked on the wind up key: WindUp and Hop. After creating the methods, students program the following: Hop should be called from WindUp and WindUp should only happen if the windup key is clicked. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-N; **5.5.1** A

UNIT 4: GETTING INTO THE STORY (EVENTS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

- Create Exploratory Module 6 Alice Worlds
- Lab 6 Alice World Projects
- Module 6 Extra Programming Problem

Guiding Questions

- What are events and how are they used in interactive computer programs?
- How would you describe the different kinds of events that enable a user to interact with computer programs in different ways?
- What is an event handler and how does the event handler allow events to call methods?
- How can you write code so that it is reused to handle more than one event?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
	Web Guides Through Alice 2, "6. Get Into the Story (events): Practice Problems"	**Instructional Activity: Module 6 Open-Ended or Group-Type Practice Programming Problems** Students create a world of their choosing that involves all kinds of events (specific to the Alice programming platform): "When the world starts," "When a specific object is clicked on," a method is called and executed. Students create methods and make some methods call others, as well as having the event handlers call methods. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-D; **4.1.1** A, B, E-G; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** A-N; **5.5.1** A

Students are challenged to create programs using the computing concepts they have learned in this course. They think computationally as they create their own interactive stories, simulations or games.

UNIT 5: THE INTERNET AND BINARY NUMBERS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 6 The Internet

Essential Understandings:

- EU 2.1, EU 2.2, EU 2.3, EU 3.1, EU 4.2, EU 6.1, EU 6.2, EU 6.3, EU 7.5

Projects and Major Assignments:

- The Internet as System and Spirit
- Binary Numbers
- Internet Protocols
- Cybersecurity Encryption
- Hackers and Hardware

Guiding Questions

- How would you explain the Internet in everyday language?
- How has the Internet shaped our current society and how may that change in the future?
- What are the pros and cons of encryption versus free speech?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 6.1.1: Explain the abstractions in the Internet and how the Internet functions. [P3] **LO 6.2.1:** Explain characteristics of the Internet and the systems built on it. [P5] **LO 6.2.2:** Explain how the characteristics of the Internet influence the systems built on it. [P4]	**Print** Abelson, Ledeen, and Lewis, appendix	**Instructional Activity: The Internet — A Peek Behind the Curtain** Students read the appendix to *Blown to Bits* for an introduction to key concepts that will be covered in greater depth throughout the unit and a broader understanding of how the Internet works. Students are broken up into small groups to discuss the following prompts: ▸ Explain in everyday language to someone without technical expertise how the Internet works. ▸ Explain the key ideas and protocols that are used to create a functioning Internet and rank them on their order of importance. **Essential knowledge addressed: 6.1.1** A-E, G; **6.2.1** A-C; **6.2.2** A-K
	Web CANVAS LMS, Google Sites, or any platform for creating and hosting Web pages or sites	**Formative Assessment: Digital Portfolio** After completing the previous activity, each student posts their group's thoughts to their online portfolio. Responses should be approximately 300 words in length. Students revise their writing over the course of the unit. **Essential knowledge addressed: 6.1.1** A-E, G; **6.2.1** A-C; **6.2.2** A-F; **6.6.2** G-K
LO 2.1.1: Describe the variety of abstractions used to represent data. [P3] **LO 2.1.2:** Explain how binary sequences are used to represent digital data. [P5]	**Web** "Count the Dots" **Supplies** Scissors, several sets of printed binary cards	**Instructional Activity: Binary Numbers — Count the Dots** Working in teams of two, students use manipulatives to learn and master the concept of binary numbers. Students are given a set of printed cards to cut out and work with, and they are shown how to count in binary numbers using the cards, flipping them over to turn the bits on and off. **Essential knowledge addressed: 2.1.1** A-G; **2.1.2** A, B, D-F

This formative assessment helps the students process some of the complex concepts introduced. Students are asked to refine their writing over the course of the unit as they gain a deeper understanding of the concepts.

It is a good idea to bring up some of the students in front of the class to model the concept before the small teams start working.

UNIT 5: THE INTERNET AND BINARY NUMBERS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 6 The Internet

Essential Understandings:

- EU 2.1, EU 2.2, EU 2.3, EU 3.1, EU 4.2, EU 6.1, EU 6.2, EU 6.3, EU 7.5

Projects and Major Assignments:

- The Internet as System and Spirit
- Binary Numbers
- Internet Protocols
- Cybersecurity Encryption
- Hackers and Hardware

Guiding Questions

- **How would you explain the Internet in everyday language?**
- **How has the Internet shaped our current society and how may that change in the future?**
- **What are the pros and cons of encryption versus free speech?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 2.1.1: Describe the variety of abstractions used to represent data. [P3] **LO 2.1.2:** Explain how binary sequences are used to represent digital data. [P5]	**Web** "Binary Bingo Practice"	**Instructional Activity: Binary Bingo** Students play Binary Bingo (a simple online game) in small groups. They are then able to transition to individual problems and a written summary that will be the formative assessment (see the next activity). **Essential knowledge addressed: 2.1.1** A-G; **2.1.2** A, B, D-F
		Formative Assessment: Binary Bingo Students summarize the concept of binary code on their online portfolio (300 words). **Essential knowledge addressed: 2.1.1** A-G; **2.1.2** A, B, D-F
LO 4.2.1: Express an algorithm in a language. [P5] **LO 6.1.1:** Explain the abstractions in the Internet and how the Internet functions. [P3] **LO 7.5.1:** Access, manage, and attribute information using effective strategies. [P1]	**Web** CSTA - Curriculum Resources, "Unit 4 - The Internet and Impact," (Lesson 2) "Evaluating Web Sites: Criteria and Tools"	**Instructional Activity: Website Mechanics** This activity has students think about how a website works. They brainstorm different types of websites that exist and what types of actions these sites can perform. I assign small groups of students a website to evaluate. They rate the website by criteria such as author, date of publication, intended audience, point of view or bias, and verifiability and are given pointers on how to spot propaganda and misinformation. Students present their findings to the class and defend their positions. **Essential knowledge addressed: 4.2.1** B; **6.1.1** A-E; **7.5.1** A-C
LO 4.2.1: Express an algorithm in a language. [P5] **LO 6.1.1:** Explain the abstractions in the Internet and how the Internet functions. [P3]		**Instructional Activity: Languages of the Internet** I give the students a short presentation on HTML, CSS, PHP, JavaScript, and SQL and how each language is used in Web development. Students then draw a Web page on a sheet of paper, including at least one example of each language. Next to each example the students write the language that is represented. **Essential knowledge addressed: 4.2.1** B; **6.1.1** A-E

Illustrations and examples are encouraged and should be included in the students' writing.

Students do not see the code that is used to create the different languages: it is more important that they understand the functionalities.

UNIT 5: THE INTERNET AND BINARY NUMBERS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 6 The Internet

Essential Understandings:
▶ EU 2.1, EU 2.2, EU 2.3, EU 3.1, EU 4.2, EU 6.1, EU 6.2, EU 6.3, EU 7.5

Projects and Major Assignments:
▶ The Internet as System and Spirit ▶ Binary Numbers ▶ Internet Protocols ▶ Cybersecurity Encryption ▶ Hackers and Hardware

Guiding Questions

▶ **How would you explain the Internet in everyday language?** ▶ **How has the Internet shaped our current society and how may that change in the future?** ▶ **What are the pros and cons of encryption versus free speech?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 4.2.1: Express an algorithm in a language. [P5] **LO 6.1.1:** Explain the abstractions in the Internet and how the Internet functions. [P3]	**Web** CSTA - CS Principles Resources, "Unit 4 - The Internet and Impact," (Lesson 2)	**Summative Assessment: Anatomy of a Web Page** Students take the "What Is A Website" test to demonstrate understanding and mastery of the concepts they've learned in this unit so far. **Essential knowledge addressed: 4.2.1** B; **6.1.1** A-E	*This summative assessment addresses the following guiding question:* ▶ How would you explain the Internet in everyday language?
LO 6.1.1: Explain the abstractions in the Internet and how the Internet functions. [P3]	**Web** CSTA - CS Principles Resources, "Unit 4 - The Internet and Impact," (Lesson 2) "Evaluating Web Sites: Criteria and Tools"	**Instructional Activity: How Do You Think the Internet Works?** Students do a quick write-up of how they believe the Internet came to exist and why it works as well as it does. They share their answers with their neighbors, and then I explain the different systems behind the Internet (tying in concepts from the reading at the start of this unit). I put the students into small groups and ask: 1. When you open a web browser and type in a URL, how does the web page actually get onto your computer? 2. Is the entire Internet stored on every computer? 3. How does your computer find Microsoft.com? 4. What prevents hackers from pretending to be websites and sending you viruses? Students then do a "History of the Internet" scavenger hunt using the worksheet in the Unit 4 materials. **Essential knowledge addressed: 6.1.1** A-E	*Using the worksheet provided in the Unit 4 materials, students conduct a search to answer the questions on the "History of the Internet." Once the students have completed the hunt and mistakes have been cleared up, I show them the PowerPoint "Website Basics" to help explain how websites function.*
	Web CSTA - CS Principles Resources, "Unit 4 - The Internet and Impact," (Lesson 2)	**Summative Assessment: Why the Internet Works** Students take the "Website Basics" quiz. **Essential knowledge addressed: 6.1.1** A-E	*This summative assessment addresses all of the guiding questions for the unit.*

UNIT 5: THE INTERNET AND BINARY NUMBERS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 6 The Internet

Essential Understandings:
▸ EU 2.1, EU 2.2, EU 2.3, EU 3.1, EU 4.2, EU 6.1, EU 6.2, EU 6.3, EU 7.5

Projects and Major Assignments:
▸ The Internet as System and Spirit ▸ Binary Numbers ▸ Internet Protocols ▸ Cybersecurity Encryption ▸ Hackers and Hardware

Guiding Questions

▸ **How would you explain the Internet in everyday language?** ▸ **How has the Internet shaped our current society and how may that change in the future?** ▸ **What are the pros and cons of encryption versus free speech?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 6.1.1: Explain the abstractions in the Internet and how the Internet functions. [P3]	**Web** CSTA - CS Principles Resources, "Unit 3 - The Internet and Impact," (Lesson 3) "How Does the Internet Work?"	**Instructional Activity: Internet and Protocols** I put the students into small groups and ask, *What enables the Internet to be so large?* Next students watch the video "How Does The Internet Work" and summarize it with their group. I review the information presented (i.e., routing, Internet Protocol, peer-to-peer versus Internet networks, cloud computing, and sharing of data) and show "The Internet" PowerPoint to help them visualize the different types of protocols. **Essential knowledge addressed: 6.1.1** A-I
	Web CSTA - CS Principles Resources, "Unit 3 - The Internet and Impact," (Lesson 3) "Andrew Blum: What Is the Internet, Really?"	**Formative Assessment: Creating Protocols** Students work in small groups to examine a series of protocols and describe the structure of the data packets and how to read the messages which they attempt to decipher. Each student then writes a paragraph on why the Internet needs a common protocol, citing an example. Next the students watch Andrew Blum's TED Talk video and complete a worksheet while watching. Students reflect and share their thoughts about the video as well as their paragraphs and worksheets. They then compare their answers, come up with the "best answers" for their group, and clearly explain their thoughts. **Essential knowledge addressed: 6.1.1** A-I
	Web CSTA - CS Principles Resources, "Unit 3 - The Internet and Impact," (Lesson 3)	**Summative Assessment: Internet Test** Students take "The Internet" test. The test asks students to identfy and describe protocols and how they function. Students are encouraged to draw pictures to help explain their answer. **Essential knowledge addressed: 6.1.1** A-I
LO 6.3.1: Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]	**Print** Abelson, Ledeen, and Lewis, chapter 5	**Instructional Activity: Cybersecurity and Encryption** This activity provides students a solid overview of the topics of cybersecurity and encryption. They read chapter 5 of *Blown to Bits* and then write a 300-word reflection on the chapter that they revise at the end of the lesson. **Essential knowledge addressed: 6.3.1** C, H-K

◂ *I offer students feedback and comments on their answers.*

◂ *This summative assessment addresses all of the guiding questions for the unit.*

BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 6 The Internet

Essential Understandings:
- EU 2.1, EU 2.2, EU 2.3, EU 3.1, EU 4.2, EU 6.1, EU 6.2, EU 6.3, EU 7.5

Projects and Major Assignments:
- The Internet as System and Spirit
- Binary Numbers
- Internet Protocols
- Cybersecurity Encryption
- Hackers and Hardware

Guiding Questions

- How would you explain the Internet in everyday language?
- How has the Internet shaped our current society and how may that change in the future?
- What are the pros and cons of encryption versus free speech?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 6.3.1: Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]	**Web** CSTA - CS Principles Resources, "Unit 3 - The Internet and Impact," (Lesson 4) "The Peruvian Coin Flip" Arora, "Introduction to Cryptography Basic Principles" "Prime Numbers and Public Key Cryptography"	**Instructional Activity: Cybersecurity and Encryption Cryptography Basics** Students are put in small groups and asked to answer the following prompts: ▶ What makes encryption work? What are the principles of encryption that make it valuable for preserving privacy of data? ▶ Explain authentication (i.e., proving who sent the message). ▶ Explain integrity (i.e., ensuring the message hasn't been altered between users/viewers). ▶ Explain nonrepudiation (i.e., proving you actually sent the message). Students then watch short YouTube videos: Introduction to Cryptography Basic Principles, and Prime Numbers and Public Key Cryptography. I give them the Peruvian Coin Flip scenario and put them in groups of three. Each group should come to its own conclusion by the end of the activity that they will share with the class. **Essential knowledge addressed: 6.3.1** C, H-K
		Formative Assessment: Encoding Activity In this assessment, students are tasked with "Securing a Quotation." They write a message on an 8 ½ x 11 piece of paper. To maximize the security of their message they write it smaller; they should notice how that takes more time to understand (smaller writing is an example of making the message a bit more secure). They work with other students to discover the best way to store the information, considering size, speed, and security. **Essential knowledge addressed: 6.3.1** C, H-K

Emphasize that encryption may make information more secure, but it takes longer to read and understand than if it was plain text.

It is important to model to how the Peruvian Coin Flip activity works. It is helpful if you have students try small sections of the activity with you before trying it on their own.

For this activity students encode data in unique encryption patterns that they devise: others shouldn't be able to directly read the papers. The patterns don't need to be covered in the unit but should work with respect to this activity. This is to help them conceptualize and think of what is meant by encoding. As they create more secure methods, students will understand that more memory is used to encrypt the information.

UNIT 5: THE INTERNET AND BINARY NUMBERS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 6 The Internet

Essential Understandings:
- EU 2.1, EU 2.2, EU 2.3, EU 3.1, EU 4.2, EU 6.1, EU 6.2, EU 6.3, EU 7.5

Projects and Major Assignments:
- The Internet as System and Spirit ▸ Binary Numbers ▸ Internet Protocols ▸ Cybersecurity Encryption ▸ Hackers and Hardware

Guiding Questions

▸ **How would you explain the Internet in everyday language?** ▸ **How has the Internet shaped our current society and how may that change in the future?** ▸ **What are the pros and cons of encryption versus free speech?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 6.3.1: Identify existing cybersecurity concerns and potential options to address these issues with the Internet and the systems built on it. [P1]	**Web** CSTA - CS Principles Resources, "Unit 3 - The Internet and Impact," (Lesson 4)	**Summative Assessment: Encryption Test** Students take the Encryption Test to measure learning. They revise/ edit their writing from the Cyber Security and Encryption activity and place it in their online portfolio. **Essential knowledge addressed: 6.3.1** C, H-K
	Web CSTA - CS Principles Resources, "Unit 3 - The Internet and Impact," (Lesson 5) Kuenning, "How Does a Computer Virus Scan Work?" Charles, "The Types of Hackers: Black Hat, White Hat or a Grey Hat Hacker, Which Type Are You?"	**Instructional Activity: Hackers, Software, and Hardware** I put students into small groups and they answer the following prompts: ▸ How does the structure of the Internet allow one to access another person's computer? ▸ Does a person always know if their computer is being accessed? ▸ What does a virus scanner do? ▸ Who are hackers? Are there any "good" hackers and what do they do? ▸ How do you protect your information? The groups then share their conclusions with the class. Next I explain how the trust model of the Internet works. I also explain how virus-scanners work and the importance of cybersecurity, the various hardware items that are used to implement cybersecurity, and the value of "white hat" hackers. **Essential knowledge addressed: 6.3.1** A-C, E, F, L, M

This summative assessment addresses all of the guiding questions for this unit.

UNIT 5: THE INTERNET AND BINARY NUMBERS

Estimated Time: 10 Hours

BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 6 The Internet

Essential Understandings:
- EU 2.1, EU 2.2, EU 2.3, EU 3.1, EU 4.2, EU 6.1, EU 6.2, EU 6.3, EU 7.5

Projects and Major Assignments:
- The Internet as System and Spirit ▸ Binary Numbers ▸ Internet Protocols ▸ Cybersecurity Encryption ▸ Hackers and Hardware

Guiding Questions

▸ How would you explain the Internet in everyday language? ▸ How has the Internet shaped our current society and how may that change in the future? ▸ What are the pros and cons of encryption versus free speech?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 6.3.1: Connect the concern of cybersecurity with the Internet and the systems built on it. [P1]	**Web** CSTA - CS Principles Resources, "Unit 3 - The Internet and Impact," (Lesson 5)	**Formative Assessment: Encrypting a Message** I put students in groups of at least three and have them do an activity to express the trust model. For example: 1. Student A writes a message, uses a color to "sign" it, folds the paper, and passes it to Student B. 2. Student B unfolds the paper and Student A's color verifies its origin. Student B then reads the message. 3. Student B writes a message and signs it. They try to pass it to Student A, but another student takes it. That student cannot unfold it because they don't know how Student A unfolds it (their private key). 4. Another student writes a message to Student A pretending to be Student B. However, they use a different color to sign it, so Student A knows it's fake. **Essential knowledge addressed: 6.3.1** A-C, E, F, L, M
		Summative Assessment: Cybersecurity Students take the "Cybersecurity" test. **Essential knowledge addressed: 6.3.1** A-J, L, M

I walk around verifying students' work and asking questions that challenge students to rethink the problem. If a group finishes early, I make the problem a bit more difficult and let them work on it while the other groups are completing the task.

This summative assessment addresses all of the guiding questions for the unit.

AP Performance Task

ESTIMATED TIME: 8 hours

Explore – Impact of Computing Innovations

Students complete the Explore Performance Task after Unit 5. By this point, students are well prepared to consider and write about a computing innovation and the impact it may have on the social, economic, and cultural areas of our lives. Students have had many opportunities to practice their writing skills by completing assignments that require them to answer questions about computing and computational thinking.

Additionally, we worked on a technology and society assignment that addressed some of the learning objectives for Big Idea 7: Global Impact, and we learned about the impact of computing on a global scale based upon the *AP Computer Science Principles Curriculum Framework*. To ensure that students have sufficient time to complete the task, I have attempted to teach students to write clearly and concisely.

Students also practiced researching topics effectively and efficiently. I have tried to encourage their curiosity and interest in computing and computational thinking. Students viewed several TED Talks on technology and will pick one that interests them for the performance task. This leads to researching information and finding different sources of information for completing the task.

Students concentrate on the written requirements as outlined by the College Board for this task. We have also discussed and created different types of visual artifacts, such as an infographic or a movie, that clearly detail the potential benefits and harmful effects of computing innovation.

UNIT 6: MATHEMATICAL EXPRESSIONS AND FUNCTIONS

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:
▶ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:
▶ Create Exploratory Module 7 Alice Worlds ▶ Labs 7 Alice World Projects

Guiding Questions

▶ What does a function do within a computer program? ▶ What is the difference between a function and a method? ▶ What are mathematical expressions and functions and how they are used in computer programs? ▶ How can functions be used to abstract complex mathematical calculations?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2]	**Web** Expeditions Through Alice, "7. Calculating Realism (mathematical expressions and functions)" Guides Through Alice 2, "7. Calculating Realism (mathematical expressions and functions): Book" (Explanation of code sections 7.2 –7.6)	**Instructional Activity: Exploratory Module 7 Alice World** Students work in pairs to create and complete the exploratory module 7 Alice world. Students use functions (purple blocks/tiles) to obtain information on objects' height, width, and depth. Using this information, students program the exact distance an object must move over another object. Students create mathematical expressions to calculate amounts they can use to control moving, turning, etc., so that objects can interact realistically based on their size and locations. They also create a function that abstracts a complex mathematical expression used to make a computer program more realistic. Students reuse this function and learn that functions are similar to methods in that they have names and take parameters. The big difference is that a function returns a value. Comments are required for these programs. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** A, B; **2.2.3** B; **2.3.1** A-C; **4.1.1** A, B, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F; **5.1.2** H-J; **5.1.3** A, B, D, F; **5.2.1** A- E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E

UNIT 6: MATHEMATICAL EXPRESSIONS AND FUNCTIONS

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:
- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:
- Create Exploratory Module 7 Alice Worlds
- Labs 7 Alice World Projects

Guiding Questions

- What does a function do within a computer program?
- What is the difference between a function and a method?
- What are mathematical expressions and functions and how they are used in computer programs?
- How can functions be used to abstract complex mathematical calculations?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 2.3.1: Use models and simulations to represent phenomena. [P3] **LO 2.3.2:** Use models and simulations to formulate, refine, and test hypotheses. [P3] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4]	**Web** Computer Science Principles Fall 2015, "Module 7 Questions Assignment" **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 7 Questions** As students are reading and following the instructions for creating the exploratory module 7 Alice world, they answer a series of questions in the module that promote a deeper understanding of the computing concepts (mathematical expressions and functions) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions are provided for this activity. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A, B, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E	*I do not find Alice intuitive for creating complex mathematical expressions or functions; students are at times confused by how to create complex mathematical expressions within the Alice platform. I work on three simple programming exercises provided in chapter 6 of Learning to Program with Alice; I lead the students through these exercises but I have them try to solve the exercises before explaining how to create and use a function.*
	Software Classroom-Response System **Web** Guides Through Alice 2, "7. Calculating Realism (mathematical expressions and functions): Lecture" ("Module 7 Calculating Realism.pptx")	**Summative Assessment: Module 7 Quiz** Using a classroom-response system, students take an assessment composed of four multiple-choice questions. **Essential knowledge addressed: 2.2.1** A; **2.2.2** B; **4.4.1** A, B, I; **4.2.1** A-C, I; **4.2.4** A; **5.5.1** A, D	*This summative assessment addresses the following guiding questions:* - What does a function do within a computer program? - What is the difference between a function and a method?

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:
- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:
- Create Exploratory Module 7 Alice Worlds
- Labs 7 Alice World Projects

Guiding Questions

- What does a function do within a computer program?
- What is the difference between a function and a method?
- What are mathematical expressions and functions and how they are used in computer programs?
- How can functions be used to abstract complex mathematical calculations?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 45-46) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.3.1, 2.3.2, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1.**	**Software** Classroom-Response System **Web** Guides Through Alice 2, "7. Calculating Realism (mathematical expressions and functions): Lecture" ("Module 7 Calculating Realism.pptx," all Demo Worlds, "Mod7LectureMaking Expressions.mov," and "Mod7Lecture-Transactive Discourse.docx")	**Formative Assessment: Module 7 Peer-Discussion Questions** Students work in groups of three to answer several multiple-choice questions using a classroom-response system. Students read the question and select an answer. After selecting their answer, they discuss it with their groups; they explain why they chose their answer and, more importantly, why they did not choose the other answers. Students then volunteer and share their responses with the class. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** A, B; **2.2.3** B; **4.1.1** A, B, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E

For this formative assessment, a document is made available to students to demonstrate that good discussions utilize transactive discourse: interactive rather than monologue discussion. I select two students, and we model transactive discourse by reading and acting out the script contained in the document. Students clearly see that utilizing transactive discourse is part of a good discussion. I share the correct answer with students and explain clearly why that answer is correct and the other ones are not.

UNIT 6: MATHEMATICAL EXPRESSIONS AND FUNCTIONS

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

▸ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4

Projects and Major Assignments:

▸ Create Exploratory Module 7 Alice Worlds ▸ Labs 7 Alice World Projects

Guiding Questions

▸ **What does a function do within a computer program?** ▸ **What is the difference between a function and a method?** ▸ **What are mathematical expressions and functions and how they are used in computer programs?** ▸ **How can functions be used to abstract complex mathematical calculations?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
All of the learning objectives from the first activities in this unit (pages 45-46) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.3.1, 2.3.2, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1.**	**Web** Guides Through Alice 2, "7. Calculating Realism (mathematical expressions and functions): Lab" (all files on page) Computer Science Principles Fall 2015, "Lab 7 Instructions" **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 7** Students continue to develop the airplane simulation Alice world from Lab 6. They create a method and event to land the plane. When users press Enter, the plane lands itself perfectly on the tarmac. Students do this by using a function to calculate how far it is to the "target" on the tarmac; then they use that value to control moving the plane to that point. Once on the tarmac, the plane moves forward until it is perfectly nose-to-nose with the stop sign. Students create and use another function to calculate how far forward to move. Students then answer a series of lab questions about events based upon the Alice program they created. Comments are required for this program. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** A, B; **2.2.3** B; **2.3.1** A-D; **2.3.2** A-H; **4.1.1** A, B, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E	◀ *Students run their code and explain to me what happens: they match the code with the executed behavior. I read their code to ensure their explanations are correct. I check the math expressions, functions, events, event handlers, and methods; students include an explanation of how abstraction was used.*
All of the learning objectives that cover Modules 2-6 are addressed.		**Summative Assessment: Modules 2-6 Test** Students take an assessment composed of 16 multiple-choice questions and one free-response question: they must write and explain their answer choice for each multiple-choice question. Two of the questions offer partial credit for selecting another answer that is not the best answer but is acceptable. Four of the multiple-choice questions require students to write a clear and concise explanation of why they chose their answer, and also why they did not choose each of the other answers. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D, **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** A, B; **2.2.3** B; **4.1.1** A, B, E-I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E	◀ *This summative assessment addresses all of the guiding questions for units 2–6.*

UNIT 7: CHOOSING YOUR PATH (IF STATEMENTS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

▶ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 3.1, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

▶ Exploratory Module 8 Alice Worlds ▶ Module 8 Questions
▶ Lab 8 Alice World Project

Guiding Questions

▶ What are If-else statements and how are they used in computer programs? ▶ What are conditional behaviors? What are some examples used in computer programs and in your daily life? ▶ How are If-else statements related to decisions and decision points in computer programs? ▶ What can If-else statements evaluate when making decisions within programs?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2]	**Web** Expeditions Through Alice, "Choosing Your Path (If statements)" Guides Through Alice 2, "8. Choosing your Path (if statements): Book" (Explanation of code sections 8.3–8.5)	**Instructional Activity: Exploratory Module 8 Alice World** Students use an If-else statement, function, parameters, and Boolean expression to evaluate and determine whether an Angel object is at a certain distance from an Egyptian object. If the distance is less than a certain number, then it evaluates to "true" and the Angel object moves a certain amount to the Egyptian object. Otherwise, it evaluates to "false" and the Angel states a string of text. Students use an If-else statement, function, parameters, and a Boolean expression to compare the height of the Angel and Egyptian; if the Angel's height is greater than the Egyptian's height, it evaluates to "true" and the Angel states "I'm taller." Otherwise, it evaluates to "false" and the Egyptian states "I'm taller." Comments are required for these programs. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D, **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-C, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G
	Web Computer Science Principles Fall 2015, "Module 8 Questions Assignment" **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 8 Questions** As students are reading and following the instructions for creating the exploratory module 8 Alice world program, they also answer a series of questions provided in module 8 that promote a deeper understanding of the computing concepts (If-else statements) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting. Step-by-step instructions for this activity are provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D, **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-C, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

Students use If-else statements (green blocks/tiles) to introduce conditional behaviors at decision points in programs (e.g., "I'll do this only if a particular condition holds"). Using If-else statements, students can decide between two different behaviors based on the value of a function: this is called "Conditional Execution." If-else statements evaluate functions via mathematical expressions or Boolean Operators — i.e. $<$, $\leq$, $>$, $\geq$, $=$ (equal), *and* $\neq$ (not equal). *Not every single line of code is executed with If-else statements.*

UNIT 7: CHOOSING YOUR PATH (IF STATEMENTS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

▸ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 3.1, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

▸ Exploratory Module 8 Alice Worlds ▸ Module 8 Questions ▸ Lab 8 Alice World Project

Guiding Questions

▸ **What are If-else statements and how are they used in computer programs?** ▸ **What are conditional behaviors? What are some examples used in computer programs and in your daily life?** ▸ **How are If-else statements related to decisions and decision points in computer programs?** ▸ **What can If-else statements evaluate when making decisions within programs?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 2.3.1: Use models and simulations to represent phenomena. [P3] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** Guides Through Alice 2, "8. Choosing your Path (if statements): Book" (Lab as Googledoc, "Mod8LabStarter.a2w," "Mod8LabPart1Complete.a2w") Computer Science Principles Fall 2015, "Lab 8 Instructions" **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 8** Students download the Mod8LabStarter.a2w Alice world. They use an If-else statement to control behavior in a conditional manner — based on behavior controlled by a random number. Students explore the differences in putting code before or after an If-else statement, compared to "inside" the If (controlled by the evaluation of the If-else statement). Students use two new object function blocks in Alice: **random number** and **ask user for a number**. Students also learn to use the **Print** command to find out the exact height of objects. Comments are required for this program. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D, **1.2.2** A; **1.2.3** A-C; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-C; **4.1.1** A-C, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.2.1** A-E; EK **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

Working in pairs, students learn other ways of introducing variation in a running program: using random behavior (objects change size, color, or move controlled by random tile functions) or user input (objects change size, color, or move controlled by a pop-up window that asks the user to choose or type something in). Students run their code and explain to me what happens — they match the code with the executed behavior. I read their code to ensure their explanations are correct.

UNIT 8: MORE COMPLEX CONTROL OF EXECUTION

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Module 9 Alice Worlds
- Module 9 Questions
- Lab 9 Alice World Project

Guiding Questions

- **How are programs created where execution is controlled by complex sets of conditions?**
- **What are nested If-else statements and compound Boolean expressions?**
- **Can you explain the complex sets of conditions that would require the use of compound Boolean expression within an If-else statement or nested If-else statements? Give examples.**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2]	**Web** Expeditions Through Alice, "9. More Complex Control of Execution: (compound Boolean expressions and nested if/elses)" Guides Through Alice 2, "9. More complex control of execution (compound Boolean expressions and nested if-elses): Book" (Explanation of code sections 9.2–9.5)	**Instructional Activity: Exploratory Module 9 Alice Worlds** Students work in pairs to create and complete the exploratory module 9 Alice worlds. Students read two different scenarios for creating two simple game programs. Both games require students to control the execution of the game by using a combination of If-else statements (single and nested) and Boolean expressions. Students create code that checks the conditions for an object to move in a certain direction when the mouse clicks (event) on certain objects and not every object. They also create code to check the conditions of the Alice world by clicking on an object (event) that randomly changes the color of objects; when the objects change to a certain combination of colors, that triggers an event for another object to "fly off" the screen. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-D; **4.1.1** A-C, E-I; **4.1.2** A-C, I; **4.2.4** A,B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

Students have a difficult time understanding the complex set of conditions (decisions) they need to consider when writing programs and using If-else statements and compound Boolean expressions. They are challenged to think clearly that computers are instructed to make decisions. It is important to tell them that they write code so that the computer can make a decision based upon something that happens within the program, such as clicking on an object that triggers an event.

UNIT 8: MORE COMPLEX CONTROL OF EXECUTION

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Module 9 Alice Worlds
- Module 9 Questions
- Lab 9 Alice World Project

Guiding Questions

- **How are programs created where execution is controlled by complex sets of conditions?**
- **What are nested If-else statements and compound Boolean expressions?**
- **Can you explain the complex sets of conditions that would require the use of compound Boolean expression within an If-else statement or nested If-else statements? Give examples.**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 2.3.1: Use models and simulations to represent phenomena. [P3] **LO 2.3.2:** Use models and simulations to formulate, refine, and test hypotheses. [P3] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** Computer Science Principles Spring 2015, "Module 9 Questions" **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 9 Questions** As students are reading and following the instructions for creating the exploratory module 9 Alice worlds, they answer a series of questions provided in the module that promote a deeper understanding of the computing concepts (If-else, nested If-else statements, and compound Boolean expressions) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions for this activity are provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-C, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G
	Software Classroom-Response System **Web** Guides Through Alice 2, "9. More complex control of execution (compound Boolean expressions and nested if-elses): Lecture" ("Modules 8 and 9 If Statements and Nested If Statements.pptx")	**Summative Assessment: Modules 8 and 9 Quiz** Using a classroom-response system, students take an assessment composed of five multiple-choice questions. **Essential knowledge addressed: 2.2.2** B; **4.1.1** A, B, I; **4.1.2** B, C, I; **4.2.1** A; **4.2.4** A; **5.5.1** D, E, G

This summative assessment addresses all of the guiding questions for the unit.

UNIT 8: MORE COMPLEX CONTROL OF EXECUTION

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Module 9 Alice Worlds
- Module 9 Questions
- Lab 9 Alice World Project

Guiding Questions

- **How are programs created where execution is controlled by complex sets of conditions?**
- **What are nested If-else statements and compound Boolean expressions?**
- **Can you explain the complex sets of conditions that would require the use of compound Boolean expression within an If-else statement or nested If-else statements? Give examples.**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 51-52) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.3.1, 2.3.2, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Software** Classroom-Response System **Web** Guides Through Alice 2, "9. More complex control of execution (compound Boolean expressions and nested if-elses): Lecture" (all files on page)	**Formative Assessment: Modules 8 and 9 Peer-Discussion Questions** Students work in groups of three to answer several multiple-choice questions using a classroom-response system. Students read the question, select an answer, and discuss it with their groups. They must explain why they chose their answer and, more importantly, why they did not choose the other answers. Students then volunteer their responses with the class. I share with them the correct answer and explain clearly why that answer is correct and the other ones are not. **Essential knowledge addressed: 1.1.1** A, B, **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** A, B; **2.2.3** B; **4.1.1** A, B, E-G, I; **4.1.2** A-C, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E

For each question, the classroom-response system is used to display histograms with recorded answers. The first histogram displays a chart with how students answer individually and a second histogram displays their answers after their group discussion. The answers are compared and recorded within the system. Students receive immediate feedback and clear explanations for both the correct and incorrect answers.

UNIT 8: MORE COMPLEX CONTROL OF EXECUTION

Estimated Time: 10–15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Module 9 Alice Worlds
- Module 9 Questions
- Lab 9 Alice World Project

Guiding Questions

- **How are programs created where execution is controlled by complex sets of conditions?**
- **What are nested If-else statements and compound Boolean expressions?**
- **Can you explain the complex sets of conditions that would require the use of compound Boolean expression within an If-else statement or nested If-else statements? Give examples.**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 51-52) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 2.3.1, 2.3.2, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Web** Guides Through Alice 2, "9. More complex control of execution (compound Boolean expressions and nested if-elses): Lab" (all files on page) **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 9** Students create a game using an Alice starter world program. They are instructed and guided to use If-else statements to control behavior in a conditional manner, either due to random movement or user-input. Students create appropriate comments to support others reading their code. The lab is separated into three parts. As students complete each part, they answer one to four technical and analytical questions based on the code they create. After completing the lab, students write two reflections explaining how their nested If-statements work in the game and how conditional statements can be used in a real-world scenario. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A-D; **2.3.2** A-H; **4.1.1** A-C, E-G, I; **4.1.2** A-C,H, I; **4.2.4** A, B; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

This lab scales back on the direct instruction given to students. Encourage them to think, "What would I do to implement that?" — perhaps writing it down for you to look over before they start coding. This lab allows students to really demonstrate their skills at knowing when to create methods, what parameters to use, etc. Really encourage them to talk to each other and to you about their ideas on how they should write the code to create the program.

UNIT 9: DOING THINGS OVER AND OVER (LOOPS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

▶ EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

▶ Create Exploratory Module 10 Alice Worlds ▶ Module 10 Questions ▶ Lab 10 Alice World Project

Guiding Questions

▶ What are iterations, and how are loops associated with iterations? ▶ What are "counted" and "nested" loops, and how they are used in computer programs? What are some examples? ▶ How are nested loops used in computer programs to create more complex behaviors or make repeated things happen in more complicated ways? ▶ How are "conditional (while)" loops and Boolean expressions used to execute a set of computing instructions?

Learning Objectives

LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2]

LO 1.2.1: Create a computational artifact for creative expression. [P2]

LO 1.2.2: Create a computational artifact using computing tools and techniques to solve a problem. [P2]

LO 1.2.3: Create a new computational artifact by combining or modifying existing artifacts. [P2]

LO 1.2.4: Collaborate in the creation of computational artifacts. [P6]

LO 1.2.5: Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4]

LO 1.3.1: Use computing tools and techniques for creative expression. [P2]

LO 2.2.1: Develop an abstraction when writing a program or creating other computational artifacts. [P2]

Materials

Web

Expeditions through Alice, "10. Doing Things Over and Over (Loops)"

Guides through Alice 2, "10. Doing Things Over and Over (Loops): Book" (Explanatory videos sections 10.2 – 10. 6)

Instructional Activities and Classroom Assessments

Instructional Activity: Exploratory Module Sections 10.1–10.4 Alice World

Students work in pairs to create and complete the exploratory module 10 Alice worlds. Students read different scenarios for creating programs that have repeating actions or iterations. Students create the Alice worlds that incorporate counted and nested loops for these iterations. Students use counted loops to execute repeating actions (or computer instructions) a specific number of times in one of the programs. The other Alice world uses nested loops to create more complex behavior within the program. Students recall and use abstraction by creating methods (procedures) for the complex set of actions (instructions) and place them in loops. Additionally, students recall and use storyboards to create these programs.

Essential knowledge addressed: **1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-H, I; **4.1.2** A-C, I; **4.2.2** A-D; **4.2.3** A-C; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

An important point to make is defining and describing iterations and using loops for iterations; I explain to students that iterations are instructions or actions that repeat within a program. Have students come up with examples of iterations in the computing programs that they use. We revisit the concept of abstraction by creating methods (procedures) for complex sets of actions. We then discuss the use of counted and nested loops as iterations and how they are used in programs.

Materials:

Web

Computer Science Principles Spring 2015, "Module 10 Questions"

Software

Google Docs or Microsoft Word

Instructional Activity: Exploratory Module Sections 10.1–10.4 Questions

As students are reading and following the instructions for creating the exploratory module 10 Alice worlds, they are also required to answer a series of questions provided in module 10 that promote a deeper understanding of the computing concepts (counted, conditional, and nested loops) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting. Step-by-step instructions are provided.

Essential knowledge addressed: **1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

UNIT 9: DOING THINGS OVER AND OVER (LOOPS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Create Exploratory Module 10 Alice Worlds
- Module 10 Questions
- Lab 10 Alice World Project

Guiding Questions

- What are iterations, and how are loops associated with iterations?
- What are "counted" and "nested" loops, and how they are used in computer programs? What are some examples?
- How are nested loops used in computer programs to create more complex behaviors or make repeated things happen in more complicated ways?
- How are "conditional (while)" loops and Boolean expressions used to execute a set of computing instructions?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 4.1.1: Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 4.2.2:** Explain the difference between solvable and unsolvable problems in computer science. [P1] **LO 4.2.3:** Explain the existence of undecidable problems in computer science. [P1] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Software** Classroom-Response System **Web** Guides Through Alice 2, "10. Doing Things Over and Over (Loops): Lecture" ("Module 10.1 – 10.4 Counted Nested Loops. pptx")	**Summative Assessment: Module Sections 10.1 to 10.4 Quiz** Using a classroom-response system, students take an assessment composed of five multiple-choice questions. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

This summative assessment addresses the following guiding questions:

- What are iterations and how are loops associated with iterations?
- What are "counted" and "nested" loops and how are they are used in computer programs? What are some examples?
- How are "nested" loops used in computer programs to create more complex behaviors or make repeated things happen in more complicated ways?

UNIT 9: DOING THINGS OVER AND OVER (LOOPS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:
- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:
- Create Exploratory Module 10 Alice Worlds
- Module 10 Questions
- Lab 10 Alice World Project

Guiding Questions

- **What are iterations, and how are loops associated with iterations?**
- **What are "counted" and "nested" loops, and how they are used in computer programs? What are some examples?**
- **How are nested loops used in computer programs to create more complex behaviors or make repeated things happen in more complicated ways?**
- **How are "conditional (while)" loops and Boolean expressions used to execute a set of computing instructions?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
All of the learning objectives from the first activities in this unit (pages 55-56) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 4.1.1, 4.1.2, 4.2.2, 4.2.3, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Software** Classroom-Response System **Web** Guides Through Alice 2, "10. Doing Things Over and Over (Loops): Lecture" ("Module 10.1 – 10.4 Counted Nested Loops. pptx," first four demo worlds, and first six demo videos)	**Formative Assessment: Module Sections 10.1–10.4 Peer-Discussion Questions** Students work in groups of three to answer several multiple-choice questions using a classroom-response system. After reading the question and selecting an answer, students are required to discuss their answers with their groups; they must clearly explain why they chose their answer and, more importantly, explain why they did not choose the other answers. Students then share their responses with the class. I share with them the correct answer and explain clearly why that answer is correct and the other ones are not. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G	*For each question, the classroom-response system is used to display histograms with recorded answers. The first histogram displays a chart with how students answer individually. A second histogram displays their answers after their group discussion. The answers are compared and recorded within the system. Students receive immediate feedback and explanations for both the correct and incorrect answers.*
	Web Expeditions Through Alice, "10. Doing Things Over and Over (Loops)" Guides Through Alice 2, "10. Doing Things Over and Over (Loops): Lecture" (Explanatory videos sections 10.4-10.6.)	**Instructional Activity: Exploratory Module Sections 10.4–10.7 Alice World** Students work in pairs to create and complete the exploratory module 10 Alice worlds. Students read different scenarios for creating programs that have repeating actions or iterations. Students incorporate conditional or while loops for iterations in these programs. They use conditional or while loops to execute and repeat code some number of times that is not known before the code runs. Students use number randomness to allow the behavior of the code to change. Students recall and use abstraction by creating methods (procedures) for the complex set of actions and placing them in loops. Additionally, students recall and use storyboards to create these programs. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G	*An important point to make for while loops is that the number of iterations for executing a set of instructions for a program is not known before running it; it may have to meet a certain condition to stop. Also discuss number randomness with students. Randomness can be useful for interactive games or other situations where it allows the behavior of the code to change each time it is run. This can be challenging for students to understand.*

UNIT 9: DOING THINGS OVER AND OVER (LOOPS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:
- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:
- Create Exploratory Module 10 Alice Worlds
- Module 10 Questions
- Lab 10 Alice World Project

Guiding Questions

▸ What are iterations, and how are loops associated with iterations? ▸ What are "counted" and "nested" loops, and how they are used in computer programs? What are some examples? ▸ How are nested loops used in computer programs to create more complex behaviors or make repeated things happen in more complicated ways? ▸ How are "conditional (while)" loops and Boolean expressions used to execute a set of computing instructions?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 55-56) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 4.1.1, 4.1.2, 4.2.2, 4.2.3, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Software** Classroom-Response System **Web** Guides Through Alice 2, "10. Doing Things Over and Over (Loops): Lecture" ("Module 10.4 – 10.7 While Loops.pptx")	**Summative Assessment: Module Sections 10.4 to 10.7 Quiz** Students take an assessment composed of four multiple-choice questions using a classroom-response system. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G
	Software Classroom-Response System **Web** Guides Through Alice 2, "10. Doing Things Over and Over (Loops): Lecture" ("Module 10.4 – 10.7 While Loops.pptx," last three demo worlds, and last four demo videos)	**Formative Assessment: Module Sections 10.4–10.7 Peer Discussion Questions** Students work in groups of three to answer several multiple-choice questions using a classroom-response system. After reading the question and selecting an answer, students are required to discuss their answers with their groups; they must clearly explain why they chose their answer and, more importantly, why they did not choose the other answers. Students then share their responses with the class. I share with them the correct answer and explain clearly why that answer is correct and the other ones are not. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

This summative assessment addresses the following guiding question:
- How are "conditional (while)" loops and Boolean expressions used to execute a set of computing instructions?

For each question, the classroom-response system is used to display histograms with recorded answers. The first histogram displays a chart with how students answer individually and a second histogram displays their answers after their group discussion. The answers are compared and recorded within the system. Students receive immediate feedback and clear explanations for both the correct and incorrect answers.

UNIT 9: DOING THINGS OVER AND OVER (LOOPS)

Estimated Time: 10 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 4.1, EU 4.2, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Create Exploratory Module 10 Alice Worlds
- Module 10 Questions
- Lab 10 Alice World Project

Guiding Questions

- **What are iterations, and how are loops associated with iterations?**
- **What are "counted" and "nested" loops, and how they are used in computer programs? What are some examples?**
- **How are nested loops used in computer programs to create more complex behaviors or make repeated things happen in more complicated ways?**
- **How are "conditional (while)" loops and Boolean expressions used to execute a set of computing instructions?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 55-56) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.3, 1.2.4, 1.2.5, 1.3.1, 2.2.1, 4.1.1, 4.1.2, 4.2.2, 4.2.3, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Web** Guides Through Alice 2, "10. Doing Things Over and Over (Loops): Lab" (all files on page) **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 10** Students create a short-story animation using an Alice starter world program. They are instructed and guided to use a loop to make an object repeat a set of actions a certain number of times. Then they use nested loops to make repeated actions happen in more complicated ways. Students create appropriate comments to support others reading their code. The lab is separated into two parts; as students complete each part, they answer one to four technical and analytical questions based upon the program (code) they create. After completing the lab, students write a written reflection explaining how nested loops prove to be useful for writing the code for the lab and provide an example of how they might use nested loops in another program. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

Again, this lab scales back on direct instruction given to students and allows them to really demonstrate their skills at knowing when to create methods, what parameters to use, etc. Encourage them to talk to one another and to you about their ideas on how they should write the code to create the animation.

UNIT 10: GROUPING ITEMS TOGETHER (LISTS)

Estimated Time: 5 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Module 11 Alice Worlds
- Module 11 Questions
- Lab 11 Alice World Project

Guiding Questions

▶ What are lists (also known as arrays) and what do they allow to be done in computer programs? ▶ How does randomness allow a group of objects to perform the same action but do it slightly differently within lists? What are some examples? ▶ How can looping be used with lists and randomness to make computer programs more interesting to use? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.3:** Create a new computational artifact by combining or modifying existing artifacts. [P2] **LO 1.2.4:** Collaborate in the creation of computational artifacts. [P6] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 1.3.1:** Use computing tools and techniques for creative expression. [P2] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2]	**Web** Expeditions through Alice, "11. Grouping Items Together (Lists)" Guides Through Alice 2, "11. Grouping Items Together (Lists): Book" (Explanatory videos for sections 11.2 –11.4)	**Instructional Activity: Exploratory Module 11 Alice World** Students work in pairs to create and complete the exploratory module 11 Alice worlds. Students read and explore how lists (arrays) work and how to use randomness to make lists interesting in programs. Students explore how they can make objects do the same thing (by using randomness the actions of the objects look more natural). Students are introduced to variables and creating them in a program. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G, H
	Web Expeditions through Alice, "11. Grouping Items Together (Lists)" **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 11 Questions** As students are reading and following the instructions for creating the exploratory module 11 Alice world programs, they also answer a series of questions throughout the module that promote a deeper understanding of the computing concepts (lists) they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions are provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

We discuss computer games as examples of how loops can be used with lists and randomness to make them more interesting. I also think it is important to inform students that they will put objects in a list and then will iterate over the list, meaning they will go through the list and have each object do something. Also point out that lists are often called arrays in other programming languages.

UNIT 10: GROUPING ITEMS TOGETHER (LISTS)

Estimated Time: 5 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Module 11 Alice Worlds
- Module 11 Questions
- Lab 11 Alice World Project

Guiding Questions

- What are lists (also known as arrays) and what do they allow to be done in computer programs?
- How does randomness allow a group of objects to perform the same action but do it slightly differently within lists? What are some examples?
- How can looping be used with lists and randomness to make computer programs more interesting to use? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 2.3.1: Use models and simulations to represent phenomena. [P3] **LO 2.3.2:** Use models and simulations to formulate, refine, and test hypotheses. [P3] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5] **LO 5.1.1:** Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Software** Classroom-Response System **Web** Guides Through Alice 2, "11. Grouping Items Together (Lists): Lecture" (Module 11 Lists.pptx")	**Summative Assessment: Module 11 Quiz** Using a classroom-response system, students take an assessment composed of three multiple-choice questions. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G, H	*This summative assessment addresses all of the guiding questions for the unit.*
	Software Classroom-Response System **Web** Guides Through Alice 2, "11. Grouping Items Together (Lists): Lecture" (all files on page)	**Formative Assessment: Module 11 Peer Discussion Questions** Students work in groups of three to answer several multiple-choice questions using a classroom-response system. Students read the question and select an answer. They then discuss their answers with their groups; they must clearly explain why they chose their answer and, more importantly, explain why they did not choose the other answers. Students then share their responses with the class. I share with them the correct answer and explain clearly why that answer is correct and the other ones are not. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G, H	*For each question, the classroom-response system is used to display histograms with recorded answers. The first histogram displays a chart with how students answer individually and a second histogram displays their answers after their group discussion. The answers are compared and recorded within the system. Students receive immediate feedback and clear explanations for both correct and incorrect answers.*

UNIT 10: GROUPING ITEMS TOGETHER (LISTS)

Estimated Time: 5 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 1.3, EU 2.2, EU 2.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Module 11 Alice Worlds
- Module 11 Questions
- Lab 11 Alice World Project

Guiding Questions

- What are lists (also known as arrays) and what do they allow to be done in computer programs?
- How does randomness allow a group of objects to perform the same action but do it slightly differently within lists? What are some examples?
- How can looping be used with lists and randomness to make computer programs more interesting to use? What are some examples?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the prior activities in this unit are addressed.	**Web** Guides Through Alice 2, "11. Grouping Items Together (Lists): Lab" (Lab as a Googledoc, "Mod-11LabStarter.a2w," "Mod11LabPart1 Stopping TheGameComplete.a2w," "Mod11Part1 WinnerComplete.a2w," and "Mod-11Part2Complete.a2w") **Software** Google Docs or Microsoft Word	**Instructional Activity: Lab 11** Students create a simulation and are instructed and guided to use while-loops, If-statements, random numbers, and lists. Students explore how to write programs based on specifications; they create a basic simulator for a biologist that requires the program to show the impact of a virus in the frog population. Students create appropriate comments to support others reading their code. Students build a cumulative project that integrates all of the programming concepts they've learned into a large program. The lab is separated into two parts. As students complete each part, they answer one to four technical and analytical questions based upon the program (code) they create. After completing the lab, students write a reflection about the project. **Essential knowledge addressed:** **1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **2.3.1** A, D; **2.3.2** A-H; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, **5.1.2** H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G, H

This lab challenges students to create a simulation program for someone; it has specifications that they need to follow. They are also challenged to use all of the programming concepts that they have learned in the course. This lab allows students to really demonstrate their skills at knowing when to create methods, what parameters to use, etc. Encourage them to talk to one another and to you about their ideas on how to code the simulation.

AP Performance Task

ESTIMATED TIME: 15 hours

Create – Applications from Ideas

Students complete the Create Performance Task after Unit 10. By this point, students are well prepared to create their own programs, and they have also practiced collaborating with one another on creating programs throughout the year (they were given opportunities to think of a program that they would like to create as early as Unit 3). Students are required to select different topics for the individual and collaborative programs in Alice. Students continually practiced creating their own worlds and writing about the process of creating them. Students also practiced creating videos of their programs, learned to print out the code for their programs, and completed several writing assignments to prepare them for the written component of the performance task.

Students are given a rubric that includes the requirements for a deliverable 1, deliverable 2, and final completed project to be submitted. This rubric is based upon the performance task requirements set forth by the College Board. For deliverable 1, students are required to indicate the topic, audience, and medium (static versus dynamic, video game, simulation) for their program. They must write a half-page synopsis of the animation/interactive game with the options and details they are going to create. Examples of a clearly to poorly written synopsis are provided. Students are then required to write an outline describing the objects, methods, and functions within the program and how they will interact with the world and each other. Once again, they are given examples of clearly to poorly written outlines.

For deliverable 2, students must turn in an implementation strategy diagram (and/or event handler descriptions). These are storyboards that they learned to create throughout the year. A clear and thorough diagram includes the necessary background scene(s) created with objects placed in initial positions; methods, including parameters, are listed as pseudocode.

The final programs are graded on programming clarity, flow control, programming documentation, concept and message, and length and complexity.

UNIT 11: SPREADSHEETS AND LARGE DATA SETS

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Modules 12 and 13 Alice Worlds
- Lab 11 Alice World Project

Guiding Questions

▶ How can you apply the concepts of programming in Alice to a computer application such as Excel or Google Sheets? ▶ What are the similarities between the terminology (i.e. parameters) and programming concepts (i.e. functions) for creating programs in Alice and spreadsheets in a spreadsheet program? ▶ What are some of the basic functions of spreadsheet programs that will increase a computer user's productivity in using those types of programs? ▶ Why are large data sets important in computing and computational thinking?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 1.1.1: Apply a creative development process when creating computational artifacts. [P2] **LO 1.2.1:** Create a computational artifact for creative expression. [P2] **LO 1.2.2:** Create a computational artifact using computing tools and techniques to solve a problem. [P2] **LO 1.2.5:** Analyze the correctness, usability, functionality, and suitability of computational artifacts. [P4] **LO 2.2.1:** Develop an abstraction when writing a program or creating other computational artifacts. [P2] **LO 3.1.1:** Find patterns and test hypotheses about digitally processed information to gain insight and knowledge. [P4] **LO 3.1.2:** Collaborate when processing information to gain insight and knowledge. [P6]	**Web** Expeditions Through Alice, "12: Intro to Spreadsheets" **Software** Google Sheets, Microsoft Excel, or spreadsheet program	**Instructional Activity: Exploratory Module 12 Spreadsheet** Students work in pairs to create and complete the exploratory module 12 spreadsheets. They read a scenario for creating a spreadsheet grade book, using the programming concepts learned in the previous units. Terminology that is used for creating programs is used for creating this spreadsheet file. If-else statements, functions, math expressions, pseudocode, and parameters are incorporated in this exercise. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **3.1.1** A, B, D, E; **3.1.2** A-C; **3.1.3** B-E; **3.2.1** C-F; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G
	Web Expeditions Through Alice, "12: Intro to Spreadsheets" Computer Science Principles Spring 2015, "Module 12 Questions Assignment" **Software** Google Docs or Microsoft Word	**Instructional Activity: Exploratory Module 12 Questions** As students are reading and following the instructions for creating the exploratory module 12 spreadsheet, they also must answer a series of questions in the module that promote a deeper understanding of the computing concepts they are learning. Students answer these questions in a shared Google Doc or in their own Microsoft Word document, using MLA formatting; step-by-step instructions are provided. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **3.1.1** A, B, D, E; **3.1.2** A-C; **3.1.3** B-E; **3.2.1** C-F; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

At first students find it very confusing to apply the programming concepts learned in Alice to spreadsheet programs. I continually make this connection. When using functions such as SUM, we make the connection with the functions used in Alice programs. Parameters are used to define the cells being added in a function, and If-else statements are used in the spreadsheet file and are connected to If-else statements used in Alice.

UNIT 11: SPREADSHEETS AND LARGE DATA SETS

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Modules 12 and 13 Alice Worlds
- Lab 11 Alice World Project

Guiding Questions

▶ **How can you apply the concepts of programming in Alice to a computer application such as Excel or Google Sheets?** ▶ **What are the similarities between the terminology (i.e. parameters) and programming concepts (i.e. functions) for creating programs in Alice and spreadsheets in a spreadsheet program?** ▶ **What are some of the basic functions of spreadsheet programs that will increase a computer user's productivity in using those types of programs?** ▶ **Why are large data sets important in computing and computational thinking?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
LO 3.1.3: Explain the insight and knowledge gained from digitally processed data by using appropriate visualizations, notations, and precise language. [P5] **LO 3.2.1:** Extract information from data to discover and explain connections or trends. [P1] **LO 3.2.2:**. Determine how large data sets impact the use of computational processes to discover information and knowledge. [P3] **LO 3.3.1:** Analyze how data representation, storage, security, and transmission of data involve computational manipulation of information. [P4] **LO 4.1.1:** Develop an algorithm for implementation in a program. [P2] **LO 4.1.2:** Express an algorithm in a language. [P5]	**Software** Classroom-Response System **Web** Guides Through Alice 2, "12. Intro to Spreadsheets): Lecture" ("Module 12 Intro Spreadsheets. pptx")	**Summative Assessment: Module 12 Quiz** Using a classroom-response system, students take an assessment composed of three multiple-choice questions. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **3.1.1** A,B,D,E; **3.1.2** A-C; **3.1.3** B-E; **3.2.1** C-F; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G	*This summative assessment addresses the following guiding questions:* ▶ How can you apply the concepts of programming in Alice to a computer application such as Excel or Google Sheets? ▶ What are some of the basic functions of spreadsheet programs that will increase a computer user's productivity in using those types of programs?
	Software Classroom-Response System, Google Sheets, or Microsoft Excel **Web** Guides Through Alice 2, "12. Intro to Spreadsheets): Lecture" ("Module 12 Intro Spreadsheets. pptx")	**Formative Assessment: Module 12 Peer-Discussion Questions** Students work in groups of three to answer several multiple-choice questions using a classroom-response system. Students read the question and select an answer. They then discuss their answers with their groups, explaining why they chose their answer and why they did not choose the other answers. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **3.1.1** A, B, D, E; **3.1.2** A-C; **3.1.3** B-E; **3.2.1** C-F; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G	*Students volunteer and share their responses with the class. I then share with them the correct answer, and we discuss why that answer is correct and why the other ones are not.*

UNIT 11: SPREADSHEETS AND LARGE DATA SETS

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Modules 12 and 13 Alice Worlds
- Lab 11 Alice World Project

Guiding Questions

- **How can you apply the concepts of programming in Alice to a computer application such as Excel or Google Sheets?**
- **What are the similarities between the terminology (i.e. parameters) and programming concepts (i.e. functions) for creating programs in Alice and spreadsheets in a spreadsheet program?**
- **What are some of the basic functions of spreadsheet programs that will increase a computer user's productivity in using those types of programs?**
- **Why are large data sets important in computing and computational thinking?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
LO 5.1.1: Develop a program for creative expression, to satisfy personal curiosity or to create new knowledge. [P2] **LO 5.1.2:** Develop a correct program to solve problems. [P2] **LO 5.1.3:** Collaborate to develop a program. [P6] **LO 5.2.1:** Explain how programs implement algorithms. [P3] **LO 5.3.1:** Use abstraction to manage complexity in programs. [P3] **LO 5.4.1:** Evaluate the correctness of a program. [P4] **LO 5.5.1:** Employ appropriate mathematical and logical concepts in programming. [P1]	**Web** Guides Through Alice 2, "12. Intro to Spreadsheets: Lab" **Web/Software** Spreadsheet and word-processing programs	**Instructional Activity: Lab 12** Students create a spreadsheet and examine data on the depictions of alcohol and tobacco consumption in children's movies. Students are required to use basic spreadsheet functions and addressing types (AutoSum, basic statistical functions, If's, and VLOOKUP). Students relate spreadsheet features to Alice programming features (If statements, methods, loops, parameters), including implementing the SUM function in Alice. The lab is separated into two parts. Part 1 calculates values and entries. Part 2 implements spreadsheet functions and programming constructs. After completing the lab, students write a reflection and describe the spreadsheet in relationship to Alice. **Essential knowledge addressed:** **1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **3.1.1** A, B, D, E; **3.1.2** A-C; **3.1.3** B-E; **3.2.1** C-F; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G
	Web Expeditions Through Alice 2, "13: Spreadsheets – Working with Large Data Sets" **Software** Spreadsheet program	**Instructional Activity: Exploratory Module 13 Spreadsheet** Students work in pairs and learn how to work with large data sets within a spreadsheet program. Students use filters and AutoFilters, locking of cells, hiding/unhiding rows/columns, Sort Data, Freeze and Unfreeze rows/columns, and "Find and Replace." **Essential knowledge addressed:** **1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **3.1.1** A, B, D, E; **3.1.2** A-C; **3.1.3** B-E; **3.2.1** C-F; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

This lab challenges students to create a spreadsheet program based upon the programming concepts learned in Alice. They are challenged to think computationally in order to collect and analyze the data in the spreadsheet. Students should work together on trying to use the functions correctly in order to find the data they are required to look for. This lab requires a lot of guidance from me.

UNIT 11: SPREADSHEETS AND LARGE DATA SETS

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Modules 12 and 13 Alice Worlds
- Lab 11 Alice World Project

Guiding Questions

▸ How can you apply the concepts of programming in Alice to a computer application such as Excel or Google Sheets? ▸ What are the similarities between the terminology (i.e. parameters) and programming concepts (i.e. functions) for creating programs in Alice and spreadsheets in a spreadsheet program? ▸ What are some of the basic functions of spreadsheet programs that will increase a computer user's productivity in using those types of programs? ▸ Why are large data sets important in computing and computational thinking?

Learning Objectives	Materials	Instructional Activities and Classroom Assessments	
All of the learning objectives from the first activities in this unit (pages 64-66) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.5, 2.2.1, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.3.1, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Software** Classroom-Response System **Web** Guides Through Alice 2, "13: Spreadsheets – Working with Large Data Sets: Lecture" ("Module 13 Spreadsheets-Large Data Sets. pptx")	**Summative Assessment: Module 13 Quiz** Students take an assessment composed of three multiple-choice questions using a classroom-response system. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.3** A-C; **1.2.4** A-F; **1.2.5** A-D; **1.3.1** D; **2.2.1** A-C; **2.2.2** B; **2.2.3** B; **3.1.1** A, B, D, E; **3.1.2** A-C; **3.1.3** B-E; **3.2.1** C-F; **4.1.1** A-I; **4.1.2** A-C, I; **4.2.4** A, B, D, E; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G	*This summative assessment addresses the following guiding questions:* ▸ How can you apply the concepts of programming in Alice to a computer application such as Excel or Google Sheets? ▸ What are some of the basic functions of spreadsheet programs that will increase a computer user's productivity in using those types of programs?
	Software Classroom-Response System, Google Sheets, or Excel **Web** Guides Through Alice 2, "13: Spreadsheets – Working with Large Data Sets: Lecture" ("Module 13 Intro Spreadsheets. pptx")	**Formative Assessment: Module 13 Peer Discussion Questions** Students work in groups of three to answer one multiple-choice question using a classroom-response system. Students read the question, select an answer, and then discuss it with their groups. They must explain clearly why they chose their answer and, more importantly, why they did not choose the other answers. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.5** A-D; **2.2.1** A, B; **2.2.2** B; **3.1.1** A, B, D, E; **3.1.2** A-C; **3.1.3** B-E; **4.1.1** A-C, E-G, I; **4.1.2** A-C, I; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G	*After discussing their answers with their groups, students volunteer and share their responses with the class. I then give them the correct answer and explain clearly why that answer is correct and why the other ones are not.*

UNIT 11: SPREADSHEETS AND LARGE DATA SETS

Estimated Time: 15 Hours

BIG IDEA 1 Creativity
BIG IDEA 2 Abstraction
BIG IDEA 3 Data and Information
BIG IDEA 4 Algorithms
BIG IDEA 5 Programming

Essential Understandings:

- EU 1.1, EU 1.2, EU 2.2, EU 3.1, EU 3.2, EU 3.3, EU 4.1, EU 5.1, EU 5.2, EU 5.3, EU 5.4, EU 5.5

Projects and Major Assignments:

- Exploratory Modules 12 and 13 Alice Worlds
- Lab 11 Alice World Project

Guiding Questions

- **How can you apply the concepts of programming in Alice to a computer application such as Excel or Google Sheets?**
- **What are the similarities between the terminology (i.e. parameters) and programming concepts (i.e. functions) for creating programs in Alice and spreadsheets in a spreadsheet program?** ▸ **What are some of the basic functions of spreadsheet programs that will increase a computer user's productivity in using those types of programs?** ▸ **Why are large data sets important in computing and computational thinking?**

Learning Objectives	Materials	Instructional Activities and Classroom Assessments
All of the learning objectives from the first activities in this unit (pages 64-66) are addressed: **1.1.1, 1.2.1, 1.2.2, 1.2.5, 2.2.1, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.3.1, 4.1.1, 4.1.2, 5.1.1, 5.1.2, 5.1.3, 5.2.1, 5.3.1, 5.4.1, 5.5.1.**	**Web** Guides Through Alice 2, "Exams" ("Final Summer 2014" and "Solutions")	**Summative Assessment: Final Exam** Students take an assessment composed of 32 multiple-choice questions. Five of these offer partial credit for selecting another answer that is not the best answer. For four of the multiple-choice questions, students must clearly and concisely explain why they chose their answer and why they didn't choose the other answers: this demonstrates whether they have a deep understanding of what they have learned. Two of the questions require students to describe how a video game uses If-else statements and how social media programs use loops. One question describes a program and shows its code; students must analyze the code and explain what they think it will do. Another question describes a scenario for a program and students must write the code for executing that program. **Essential knowledge addressed: 1.1.1** A, B; **1.2.1** A-D; **1.2.2** A; **1.2.5** A-D; **2.2.1** A, B; **2.2.2** B; **3.1.1** A, B, D, E; **3.1.2** A-C; **3.1.3** B-E; **4.1.1** A-C, E-G, I; **4.1.2** A-C, I; **5.1.1** A, B, D; **5.1.2** A-F, H-J; **5.1.3** A, B, D, F; **5.2.1** A-E; **5.3.1** A-G; **5.4.1** C-F, H-N; **5.5.1** A, D, E, G

This summative assessment addresses all of the guiding questions for Units 6-11.

Resources

All links to online resources were verified before publication. In cases where links are no longer working, we suggest that you try to find the resource by doing a keyword Web search.

General Resources

"Abelson, Hal, Ken Ledeen, and Harry Lewis. *Blown to Bits: Your Life, Liberty, and Happiness after the Digital Explosion*. Addison-Wesley, 2008. Accessed November 7, 2014. **http://www.bitsbook.com/thebook**.

Alice (Version 2.4) [Software]. Carnegie Mellon University, 1999. **http://www.alice.org/index.php?page5downloads/download_alice2.4**.

Computer Science Unplugged. Michael Fellows, Tim Bell, and Ian Witten Accessed November 8, 2014. **http://csunplugged.org/activities**.

"Curriculum." Exploring Computer Science. Joanna Goode and Gail Chapman. Accessed November 5, 2014. **http://www.exploringcs.org/curriculum**.

"CSTA – CS Principles Resources." Computer Science Teachers Association/ Association for Computing Machinery (ACM). Accessed December 12, 2014. **http://csta.acm.org/Curriculum/sub/CurrResources.html**.

Dann, Wanda P., Stephen Cooper, and Randy Pausch. *Learning to Program with Alice: Custom Edition for University of California, San Diego*. Boston: Prentice Hall, 2009.

Expeditions Through Alice. Quintin Cutts, Sarah Esper, and Beth Simon. Accessed November 5, 2014. **https://sites.google.com/a/eng.ucsd.edu/expeditions-through-alice/home**.

Guides Through Alice 2. Quintin Cutts, Sarah Esper, and Beth Simon. Accessed November 5, 2014. **https://sites.google.com/a/eng.ucsd.edu/guides-through-alice-2/home**.

Lopez, Arthur. Computer Science Principles Fall 2015 and Computer Science Principles Spring 2015. Accessed July 25, 2015. **https://canvas.instructure.com/courses/944870** (Fall Semester); **https://canvas.instructure.com/courses/943888** (Spring Semester).

"Pair Programming-in-a-Box: The Power of Collaborative Learning." National Center for Women & Information Technology. Accessed November 5, 2014. **http://www.ncwit.org/resources/pair-programming-box-power-collaborative-learning**.

"Repository for Alice Materials." Susan Rodger. Duke University Department of Computer Science. Accessed November 5, 2014. **http://www.cs.duke.edu/csed/alice09**.

Unit 1 (Introduction to Computer Science Principles) Resources

"Alice Tutorials Summers 2008–2014." Susan Rodger. Duke University Department of Computer Science. Accessed July 25, 2015. **http://www.cs.duke.edu/csed/alice09/tutorials.php**.

"Growing Up Online." Frontline. PBS. Accessed November 5, 2014. **http://www.pbs.org/wgbh/pages/frontline/kidsonline**.

Guides Through Alice. "Discussion Videos." Quintin Cutts, Sarah Esper, and Beth Simon. Accessed November 5, 2014. **https://sites.google.com/a/eng.ucsd.edu/guidesthroughalice/discussion-videos**.

"Planning with Pseudocode." Khan Academy. Accessed June 5, 2015. **https://www.khanacademy.org/computing/computer-programming/programming/good-practices/p/planning-with-pseudo-code**.

Unit 2 (Telling and Dividing a Story [Methods]) Resources

No unit-specific resources.

Unit 3 (Behaving the Same vs Differently [Parameters]) Resources

Ted.com. Accessed November 5, 2014. **http://www.ted.com/talks?topics[]=technology&sort=newest**.

Unit 4 (Get into the Story [Events]) Resources

No unit-specific resources.

Unit 5 (The Internet and Binary Numbers) Resources

Arora, Himanshu. "Introduction to Cryptography Basic Principles." The Geek Stuff. Accessed November 8, 2014. **http://www.thegeekstuff.com/2012/07/cryptography-basics**.

"Andrew Blum: What Is the Internet, Really?" TED Talks. Youtube. Video, 11:59. **https://youtu.be/XE_FPEFpHt4**.

"Binary Bingo Practice." Kevin's Number Conversion Practice through Bingo. Kevin James. **http://courses.cs.vt.edu/~cs1104/Bingo/bingo.kevin.html**.

Charles, Kellep. "The Types of Hackers: Black Hat, White Hat or a Grey Hat Hacker, Which Type are you?" Examiner.com. Accessed November 8, 2014. **http://www.examiner.com/article/the-types-of-hackers-black-hat-white-hat-or-a-grey-hat-hacker-which-type-are-you**.

"Count the Dots." Computer Science Unplugged. Fellows, Michael, Tim Bell, and Ian Witten. Accessed November 8, 2014. **http://csunplugged.org/binary-numbers/#Count_the_Dots**.

"CSTA – CS Principles Resources." Computer Science Teachers Association/ Association for Computing Machinery (ACM). Accessed 12 December 2014. **http://csta.acm.org/Curriculum/sub/CurrResources.html**.

"Evaluating Web Sites: Criteria and Tools." Cornell University Library. **https://olinuris.library.cornell.edu/ref/research/webeval.html**.

Resources (continued)

"How Does the Internet Work?" Data Centers Canada: Colocation and Disaster Recovery Facilities. Youtube. Video, 5:37. Accessed November 8, 2014. **https://youtu.be/i5oe63pOhLI**.

Kuenning, Geoff. "How does a computer virus scan work?" Scientific American. Accessed November 8, 2014. **http://www.scientificamerican.com/article/how-does-a-computer-virus**.

"The Peruvian Coin Flip." Computer Science Unplugged. Fellows, Michael, Tim Bell, and Ian Witten. Accessed November 8, 2014. **http://csunplugged.org/cryptographic-protocols**.

"Prime Numbers & Public Key Cryptography." Simon Pampena. Youtube. Video, 2:57. Accessed November 8, 2014. **https://www.youtube.com/watch?v=56fa8Jz-FQQ**.

Unit 6 (Mathematical Expressions and Functions) Resources

No unit-specific resources.

Unit 7 (Choosing Your Path [If Statements]) Resources

No unit-specific resources.

Unit 8 (More Complex Control of Execution) Resources

No unit-specific resources.

Unit 9 (Doing Things Over and Over [Loops)] Resources

No unit-specific resources.

Unit 10 (Grouping Items Together [Lists]) Resources

No unit-specific resources.

Unit 11 (Spreadsheets and Large Data Sets) Resources

No unit-specific resources.