CS 51 Computer Science Principles

APCSP Module 3: Data, Internet, Computer and Programming

Unit 1: Creative Development

SYLLABUS

DR. ERIC CHOU

IEEE SENIOR MEMBER

Eric Chou, Ph.D.

Adjunct Professor, Lewis Univ.

IEEE Senior Member
CSTA Member

IEEE

USC

National Taiwan University

Curriculum Overview and Goals

Computing affects almost all aspects of modern life and *all* students deserve access to a computing education that prepares them to pursue the wide array of intellectual and career opportunities that computing has made possible. Here is a brief summary of each of the units in the Code.org CSP curriculum.

Unit 1: The Internet	Learn how the multi-layered systems of the Internet function as you collaboratively solve problems and puzzles about encoding and transmitting data, both 'unplugged' and using Code.org's Internet Simulator.
Unit 2: Digital Information	Use a variety of digital tools to look at, generate, clean, and manipulate data to explore the relationship between information and data. Create and use visualizations to identify patterns and trends.
Unit 3: Algorithms and programming	Learn the JavaScript language with turtle programming in Code.org's App Lab. Learn general principles of algorithms and program design that are applicable to any programming language.
Unit 4: Big Data and Privacy	Research current events around the complex questions related to public policy, law, ethics, and societal impact. Learn the basics of how and why modern encryption works.
Unit 5: Building Apps	Continue learning how to program in the JavaScript language. Use Code.org's App Lab environment to create a series of applications that live on the web. Each app highlights a core concept of programming.

This course is not a tour of current events and technologies. Rather, it seeks to provide students with a "future proof" foundation in computing principles so that they are adequately prepared with both the knowledge and skills to live and meaningfully participate in our increasingly digital society, economy, and culture.

BIG IDEA 1: CREATIVE DEVELOPMENT (CRD)

When developing computing innovations, developers can use a formal, iterative design process or a less rigid process of experimentation. While using either approach, developers will encounter phases of investigating and reflecting, designing, prototyping, and testing. Additionally, collaboration is an important tool at any phase of development, because considering multiple perspectives allows for improvement of innovations.

BIG IDEA 2: DATA (DAT)

Data are central to computing innovations because they communicate initial conditions to programs and represent new knowledge. Computers consume data, transform data, and produce new data, allowing users to create new information or knowledge to solve problems through the interpretation of those data. Computers store data digitally, which means that the data must be manipulated in order to be presented in a useful way to the user.

BIG IDEA 3: ALGORITHMS AND PROGRAMMING (AAP)

Programmers integrate algorithms and abstraction to create programs for creative purposes and to solve problems. Using multiple program statements in a specified order, making decisions, and repeating the same process multiple times are the building blocks of programs. Incorporating elements of abstraction—by breaking problems down into interacting pieces, each with their own purpose—makes writing complex programs easier. Programmers need to think algorithmically and use abstraction to define and interpret processes that are used in a program.

BIG IDEA 4: COMPUTING SYSTEMS AND NETWORKS (CSN)

Computer systems and networks are used to transfer data. One of the largest and most commonly used networks is the Internet. Through a series of protocols, the Internet can be used to send and receive information and ideas throughout the world. Transferring and processing information can be slow when done on a single computer, but leveraging multiple computers to do the work at the same time can significantly shorten the time it takes to complete tasks or solve problems.

BIG IDEA 5: IMPACT OF COMPUTING (IOC)

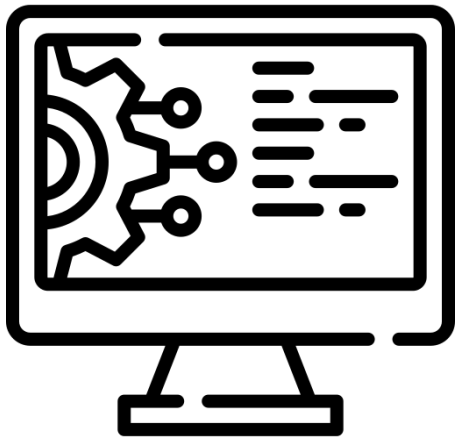
Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand the potential impacts of our programs and be responsible for the consequences. As computer users, we need to understand any potential beneficial or harmful effects and how to protect ourselves and our privacy when using a computer.

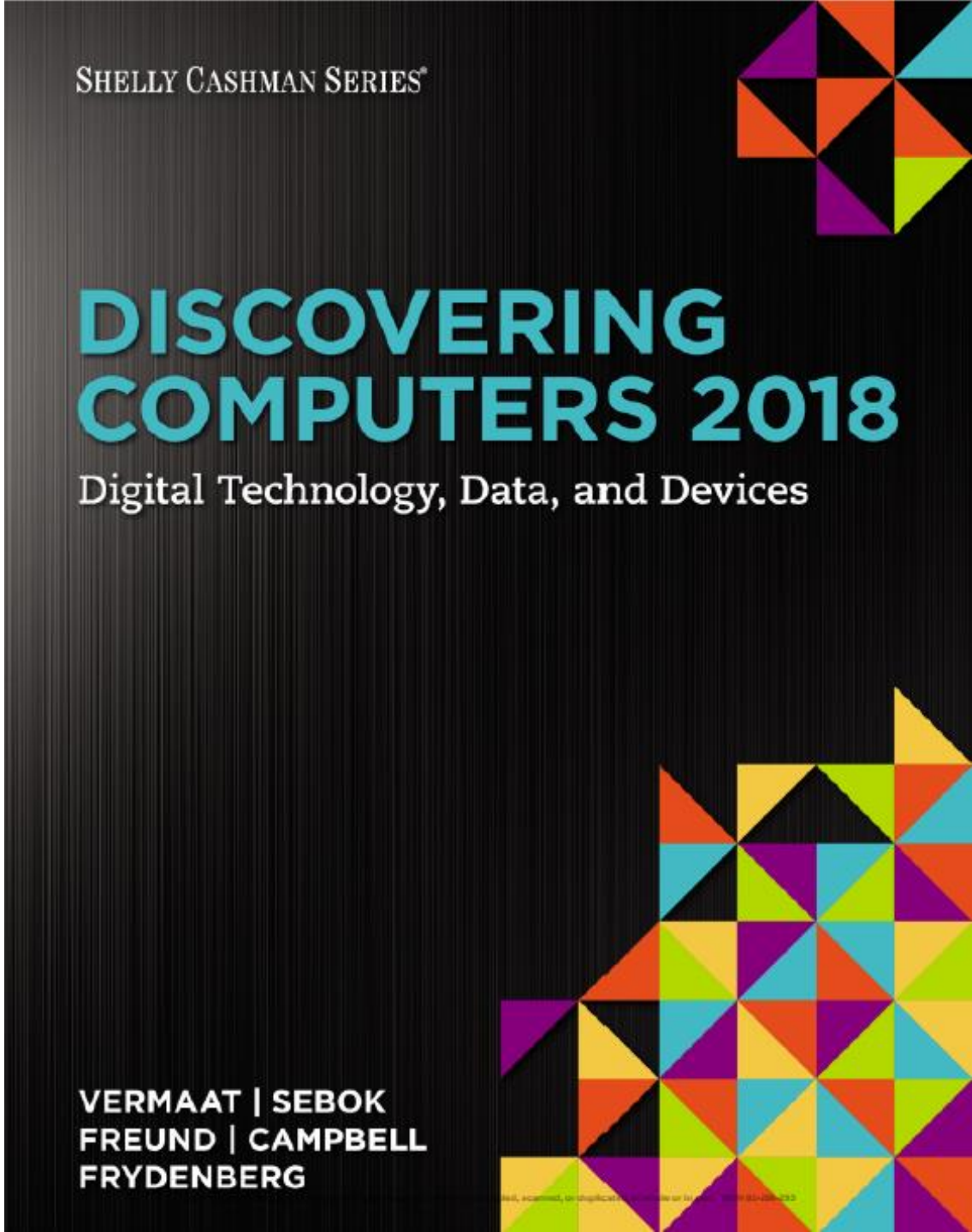
Big Ideas	Exam Weighting
Big Idea 1: Creative Development	**10–13%**
Big Idea 2: Data	**17–22%**
Big Idea 3: Algorithms and Programming	**30–35%**
Big Idea 4: Computer Systems and Networks	**11–15%**
Big Idea 5: Impact of Computing	**21–26%**

Discovering Computers

16th Edition

By Misty E. Vermaat, Susan L. Sebok, et al.





APCSP Exam Format

FALL 2023 UPDATED

Section	Question Type/Component	Number of Questions	Exam Weighting	Timing
I	Multiple-choice questions	**70**	**70%**	**120 minutes End-of-course AP Exam**
	Single-select	57		
	Single-select with reading passage about a computing innovation	5		
	Multi-select	8		
II	Create Performance Task	***See Below***	**30%**	***See Below***
	Program code, video, and Personalized Project Reference			At least 9 hours in class
	Written response related to the Create performance task	4		60 minutes End-of-course AP Exam

Video, Program Requirements, and Written Response 1 — 3 points

General Scoring Notes

- Responses should be evaluated solely on the rationale provided.
- Responses must demonstrate all criteria, including those within bulleted lists, in each row to earn the point for that row.
- Terms and phrases defined in the terminology list are italicized when they first appear in the scoring criteria.

Reporting Category	Scoring Criteria	Decision Rules
Course Project: Video **(0–1 points)**	The video demonstrates the running of the program including: • *input* • *program functionality* • *output*	**Consider only the video when scoring this point.** **Do NOT award a point if the following is true:** • The video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited).
Course Project: Program Requirements **(0–1 points)**	The program code includes: • *A list* • *A procedure* • *A call to the procedure* • *Selection* • *Iteration*	**Consider the Personalized Project Reference (or Program Code if necessary) when scoring this point.** • If the program requirements do not appear in the Personalized Project Reference, consider the full program code file when scoring this point. • The procedure does not need to have a parameter to earn this point. • The code segments demonstrating selection and iteration do not need to appear in the same algorithm to earn this point. • The code segments demonstrating selection and iteration do not need to be contained in a procedure to earn this point. **Do NOT award a point if any one or more of the following is true:** • The list is a one-element list. • The use of the list is irrelevant or not used in the program. The use of either the selection or the iteration is trivial (i.e., does not affect the outcome of the program).

Reporting Category	Scoring Criteria	Decision Rules
Written Response 1: Program Function and Purpose **(0–1 points)**	The written response: • identifies an expected user of the program. • describes one way the program's design meets the needs of the identified user.	**Consider Written Response 1 and the student's Program Code when scoring this point.** • The response must relate the program design to the needs of the identified user. **Do NOT award a point if the following is true:** • The description of the design is implausible, inaccurate, or inconsistent with the program.

Written Response 2 — 3 points

General Scoring Notes

- Responses should be evaluated solely on the rationale provided.
- Responses must demonstrate all criteria, including those within bulleted lists, in each row to earn the point for that row.
- Terms and phrases defined in the terminology list are italicized when they first appear in the scoring criteria.

Reporting Category	Scoring Criteria	Decision Rules
Written Response 2(a): Algorithm Development **(0–1 points)**	The written response: • identifies the number of times the body of the iteration statement will execute. AND ○ describes a condition or error that would cause an infinite loop. OR ○ if no such condition or error exists, explains how the loop could be modified to cause an infinite loop.	**Consider the Personalized Project Reference and Written Response 2(a) when scoring this point.** • If multiple iteration statements are included in the Procedure section of the Personalized Project Reference, use the first iteration statement to determine whether the point is earned. • The iteration statement does not need to be contained in a procedure to earn this point. **Do NOT award a point if the following is true:** • The identified number of times the body of the iteration statement will execute does not match the code.
Written Response 2(b): Errors and Testing **(0–1 points)**	The written response: • describes a change to the procedure that will result in a run-time error. • explains why the change will result in a run-time error.	**Consider the Personalized Project Reference and Written Response 2(b) when scoring this point.** • If multiple procedures are included in part (i) of the Procedure section of the Personalized Project Reference, use the first procedure to determine whether the point is earned. **Do NOT award a point if any one or more of the following is true:** • A procedure is not identified in part (i) of the Procedure section of the Personalized Project Reference. • The response does not apply to the procedure in in part (i) of the Procedure section of the Personalized Project Reference. • The response describes expected behavior that is implausible, inaccurate, or inconsistent with the program.

Reporting Category	Scoring Criteria	Decision Rules
Written Response 2(c): Data and Procedural Abstraction **(0–1 points)**	The written response: • explains in detailed steps an algorithm that uses `isEqual` to count the number of elements in the list that are equal to a certain value.	**Consider the Personalized Project Reference and Written Response 2(c) when scoring this point.** • If multiple lists are included in the List section of the Personalized Project Reference, use the first list to determine whether the point is earned. • The algorithm can be described in code, pseudocode, as a sequence of steps in English, or as a paragraph in English. • The algorithm must describe iterating over all elements of the list, calling `isEqual` on each list element and maintaining a count of the number of list elements equal to a certain value. • The algorithm must correctly determine the number of elements that are equal to a certain value. Some ways this can be determined is by storing the value in a variable, returning it, or displaying it to the user. **Do NOT award a point if the following is true:** • A list is not identified in the List section of the Personalized Project Reference. • The description of the algorithm is not detailed enough for someone else to write the code.

Creative Development

1.1 Collaboration

1.2 Program Function and Purpose

1.3 Program Design and Development

1.4 Identifying and Correcting Errors

Data

2.1 Binary Numbers

2.2 Data Compression

2.3 Extracting Information from Data

2.4 Using Programs with Data

Algorithms and Programming

3.1 Variables and Assignments
3.2 Data Abstraction
3.3 Mathematical Expressions
3.4 Strings
3.5 Boolean Expressions
3.6 Conditionals
3.7 Nested Conditionals
3.8 Iteration
3.9 Developing Algorithms
3.10 Lists
3.11 Binary Search
3.12 Calling Procedures
3.13 Developing Procedures
3.14 Libraries
3.15 Random Values
3.16 Simulations
3.17 Algorithmic Efficiency
3.18 Undecidable Problems

Computer Systems and Networks

4.1 The Internet

4.2 Fault Tolerance

4.3 Parallel and Distributed Computing

Impact of Computing

5.1 Beneficial and Harmful Effects

5.2 Digital Divide

5.3 Computing Bias

5.4 Crowdsourcing

5.5 Legal and Ethical Concerns

5.6 Safe Computing

CS 51 Course Schedule Spring 2024				
Date	Topic	Quiz	Practice Chapters	Practice Exam
W1-01/07	Lecture 1 Creative Development (Unit 1)	X	Barron's Unit 1	
W2-01/14	Lecture 2 Data Types (Uni 2)	X	Barron's Unit 2	
W3-01/21	Lecture 3 Data Compression (Unit 2)	X		
W4-01/28	Lecture 4 Information (Unit 2)	X	Barron's Unit 4	
W5-02/04	Lecture 5 Computer and Network Hardware	X		
W6-02/11	Lecture 6 Internet (Layer 4-7) (Unit 4)	X	Barron's Unit 5	
W7-02/18	Lecture 7 Global Impacts (Unit 5)	X		
W8-02/25	Lecture 8 Ethics and Security (Unit 5)	X		
W9-03/04	Lecture 9 App Lab Overview (Unit 3)		Barron's Unit 3 Part 1	
W10-03/11	Lecture 10 Program Structure (Unit 3)			
W11-03/18	Lecture 11 List and Loop (Unit 3)		Barron's Unit 3 Part 2	College Board Sample Questions
W12-03/25	Lecture 12 Algorithm and Library (Unit 3)			Barron Diag
W13-04/01	Lecture 13 Hackathon (Unit 3)			Barron 1
W14-04/08	AP Create Task Preparation			Barron 2
W15-04/15	APCSP College Board 2016 Discussion			APCSP College Board 2016
W16-04/22	APCSP College Board 2018 Discussion			APCSP College Board 2018
W17-04/29	APCSP College Board 2020 Discussion			APCSP College Board 2020
Total				