ECEP Expanding Computing Education Pathways

Connecticut’s Computer Science Course Reporting Guidance

Creating Common Understanding and Metrics

CSDE

CONNECTICUT STATE DEPARTMENT OF EDUCATION

Welcome

The team:

- Susan Auchincloss
- Danielle Bousquet
- Jackie Corricelli
- Seth Freeman
- Jennifer Michalek
- Chinma Uche

This project was funded by Expanding Computing Education Pathways (ECEP)

We would like to thank the following school districts for helping with this guide:

- Bridgeport
- CREC
- New Haven
- New London
- Newtown

Why a new Computer Science Course Code Guide

Improve and enhance the data collection systems and processes

Better articulate broadening participation in computer science (BPC) goals as outlined in the State Computer Science Plan

Assess the effectiveness of BPC initiatives to expand capacity, access, participation and experience of underrepresented minority students and girls in CS.

The Process of Developing the New Guide

- Utilized current data systems to understand computer science reporting trends and accuracy.
- Partnered with a variety of school districts to understand challenges of computer science course reporting
- Prepared CS Course Reporting Guidance based upon New England Metrics Project and with input from school districts
- Obtained feedback from districts

Anticipated Impact

- Understand the challenges and barriers to accurate CS reporting
- Engage with districts on course reporting expectations
- Greater awareness on the part of districts on the importance of accurate reporting
- Increased understanding by school districts on which elements constitute a computer science course

Computer Science State Plan

- Current Landscape and Strategic Goals
  - Landscape Report
  - Strategic Goals
- Diversity
- Teacher Pipeline
  - Professional Learning
  - Certification and Licensure
  - Preservice Programs
- Curriculum and Courses
  - Standards
  - Curriculum
  - Graduation Requirements
- Outreach
- Funding

Goals

- Provide common understanding of computer science
- Promote accurate course reporting
- Enable measurement of impact from computer science initiatives
- Identify trends in Connecticut computer science education

CS Cross Endorsement

Option 1

A major in computer science or at least 30 semester hours of credit related to computer science including study in each of the following areas:

- Computing Systems
- Networks and the Internet
- Data and Analysis
- Algorithms and Programming
- Impacts of Computing,

Note: At least 12 semester hours of credit must be specific to computer science.

Option 2

Subject Area Assessment:

Passing score on Praxis II

Computer Science Test #5652

Qualifying Score of 149

https://www.ets.org/praxis/prepare/materials/5652

Applying for cross-endorsement:

Requirements

Process

Certification

When required for computer education

A certificate shall be required for anyone serving in the employ of a board of education as a teacher of computer technology, computer literacy, computer programming or electronics, data processing, or related courses.

Endorsement requirements

Courses shall be taught by teachers holding appropriate grade-level certification.

- Assignment data ensures compliance between a teachers assignment and what their endorsement authorizes them to teach

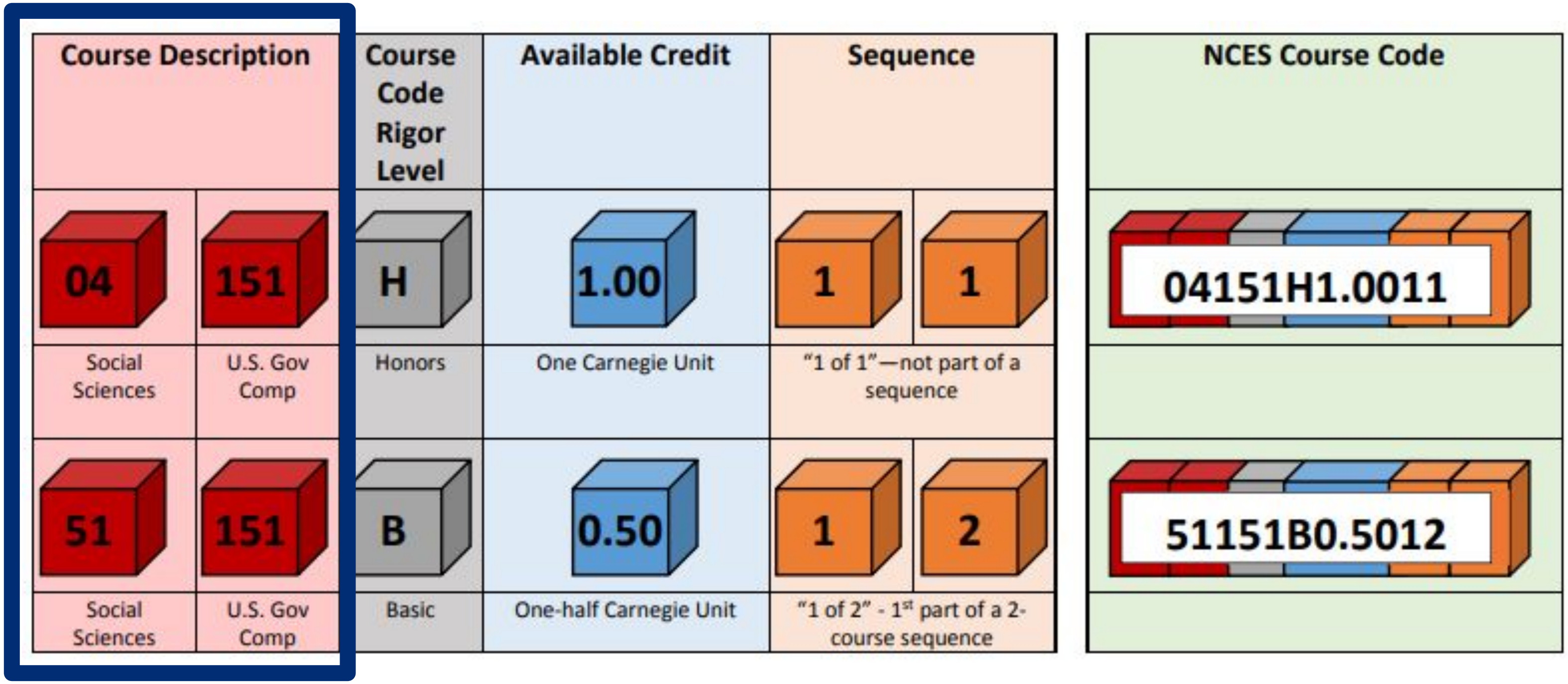
Credit

- The awarding of credits for specific courses is determined at the local level
- A teachers certification does not dictate the type of credit that can be awarded
- A teachers certification determines what they are authorized to teach and that must align to their assignment
- Class of 2023 requirements include 9 credits in science, technology, engineering and mathematics - include computer science

Using Guide with TCS

- Works with the existing Teacher-Course-Student (TCS) data collection
  - Changes to NCES/SCED versions 7.0 and 8.0 included
  - Courses will only be counted as a computer science course if they appear in this guide
  - For more information on the TCS data collection, please see the TCS help site.

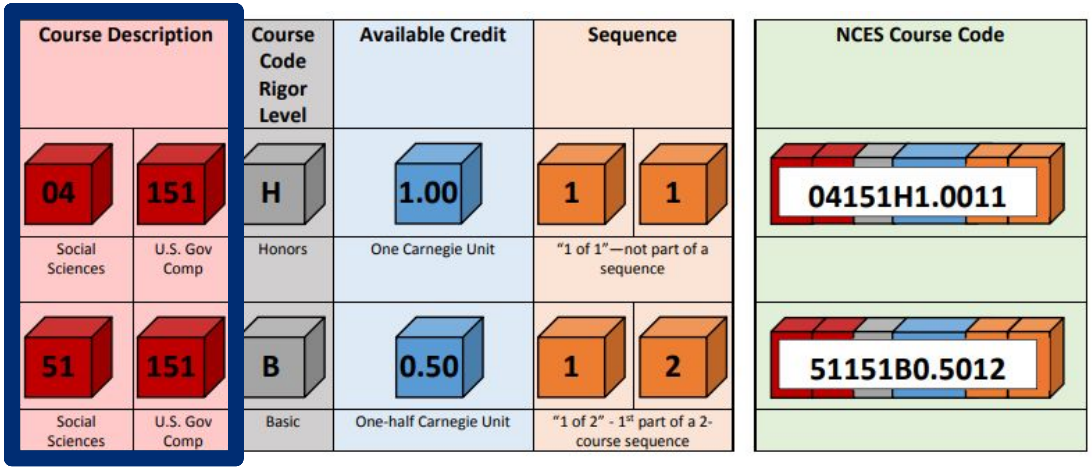
Using Guide with TCS



Course Description		Course Code Rigor Level	Available Credit	Sequence		NCES Course Code
04	151	H	1.00	1	1	04151H1.0011
Social Sciences	U.S. Gov Comp	Honors	One Carnegie Unit	"1 of 1"—not part of a sequence		
51	151	B	0.50	1	2	51151B0.5012
Social Sciences	U.S. Gov Comp	Basic	One-half Carnegie Unit	"1 of 2" - 1st part of a 2-course sequence		

Guide provides assistance with selecting course description codes that will be reported as Computer Science Courses.

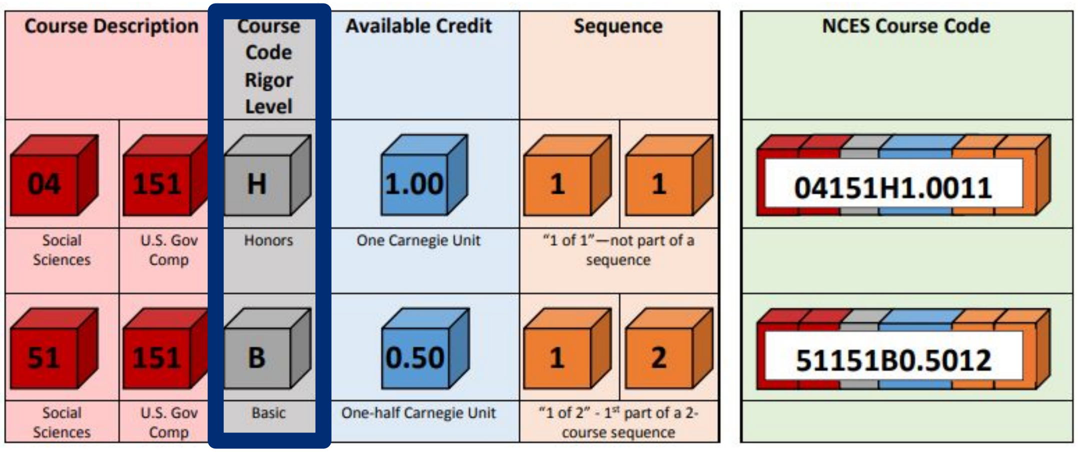
Using Guide with TCS



Course Description		Course Code Rigor Level	Available Credit	Sequence		NCES Course Code
04	151	H	1.00	1	1	04151H1.0011
Social Sciences	U.S. Gov Comp	Honors	One Carnegie Unit	"1 of 1"—not part of a sequence		
51	151	B	0.50	1	2	51151B0.5012
Social Sciences	U.S. Gov Comp	Basic	One-half Carnegie Unit	"1 of 2" - 1st part of a 2-course sequence		

Not all computer science courses fall under the Information Technology (IT) subject area (#10), and not all courses in the IT subject area are computer science courses.

Using Guide with TCS



Course Description		Course Code Rigor Level	Available Credit	Sequence		NCES Course Code
04	151	H	1.00	1	1	04151H1.0011
Social Sciences	U.S. Gov Comp	Honors	One Carnegie Unit	"1 of 1"—not part of a sequence		
51	151	B	0.50	1	2	51151B0.5012
Social Sciences	U.S. Gov Comp	Basic	One-half Carnegie Unit	"1 of 2" - 1st part of a 2-course sequence		

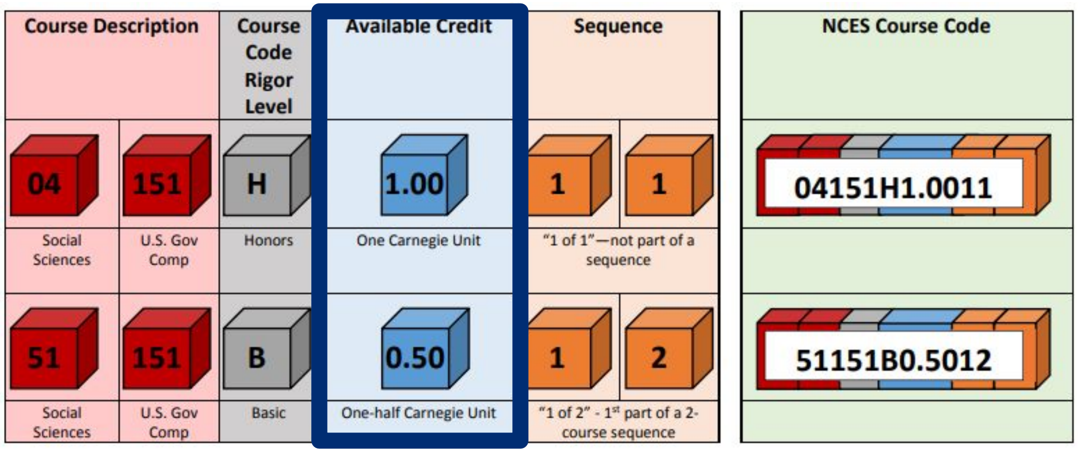
Computer science courses can be at any course rigor level, so long as the description still applies.

B=Basic G=General

E=Enriched/Advanced

H=Honors C=College-Level

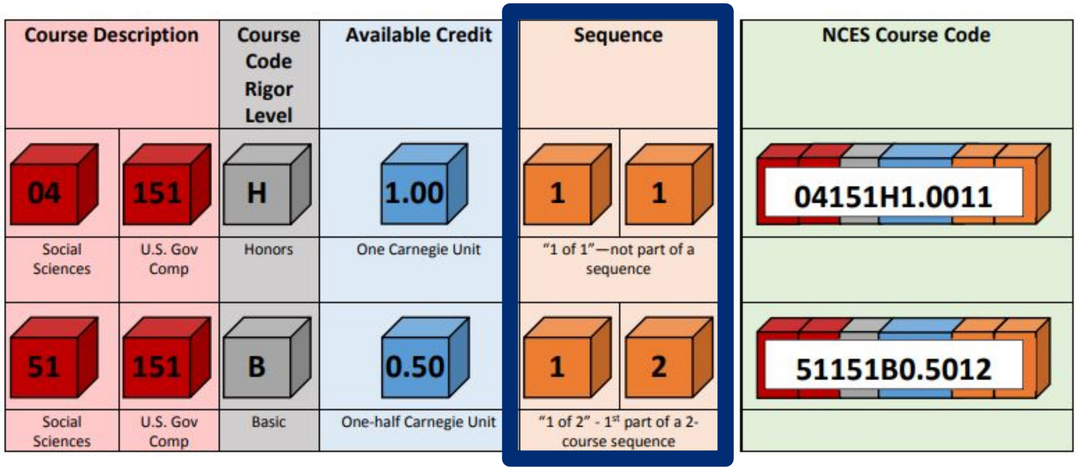
Using Guide with TCS



Course Description		Course Code Rigor Level	Available Credit	Sequence		NCES Course Code
04	151	H	1.00	1	1	04151H1.0011
Social Sciences	U.S. Gov Comp	Honors	One Carnegie Unit	"1 of 1"—not part of a sequence		
51	151	B	0.50	1	2	51151B0.5012
Social Sciences	U.S. Gov Comp	Basic	One-half Carnegie Unit	"1 of 2" - 1st part of a 2-course sequence		

Computer science courses do not have a minimum credit requirement, however they will not be reported as a Career and Technical Education (CTE) course with less than 0.5 credits.

Using Guide with TCS



Course Description		Course Code Rigor Level	Available Credit	Sequence		NCES Course Code
04	151	H	1.00	1	1	04151H1.0011
Social Sciences	U.S. Gov Comp	Honors	One Carnegie Unit	"1 of 1"—not part of a sequence		
51	151	B	0.50	1	2	51151B0.5012
Social Sciences	U.S. Gov Comp	Basic	One-half Carnegie Unit	"1 of 2" - 1st part of a 2-course sequence		

Course Sequence can be used to differentiate two courses with the same course description code that are intended to be taken following one another in cases where two different course codes could not be found.

> *Note: If the courses in the sequence are taken concurrently, they will not register as different courses.*

Using Guide with TCS

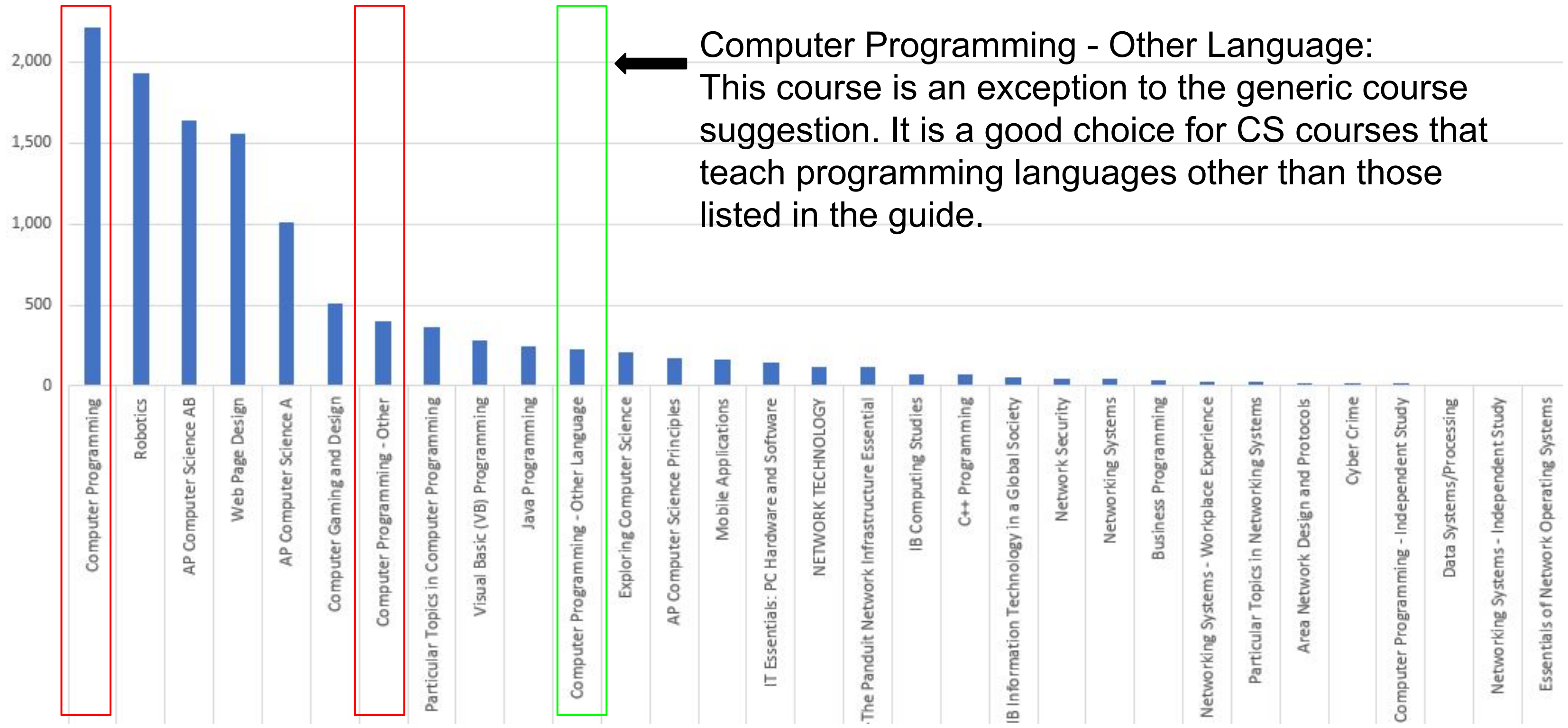
Example: Java Programming (10155)

- Java vs. Advanced Java
  - If Java is a prerequisite for Advanced Java, use the course sequence to differentiate.
    - Java: 10155G1.0012
    - Advanced Java: 10155G1.0022
  - If the courses cover most of the same material, but one is intended for more advanced students, use the rigor code to differentiate.
    - Java: 10155G1.0011
    - Advanced Java: 10155E1.0011
    - Honors Java: 10155H1.0011

Expanding Computing Education Pathways

Generic Course Codes

Whenever possible, please try to avoid using generic codes such as 10199: Computer Programming - Other



Computer Programming - Other Language: This course is an exception to the generic course suggestion. It is a good choice for CS courses that teach programming languages other than those listed in the guide.

2019-20 Course by Subject Data, available on edsight.ct.gov

CSDE

Organization of Courses

The CT CS Course Reporting Guide organizes CS Courses into two GROUPS.

Group 1 Courses:

- Align to CSTA Standards
- Would be difficult to teach and not align to CSTA Standards

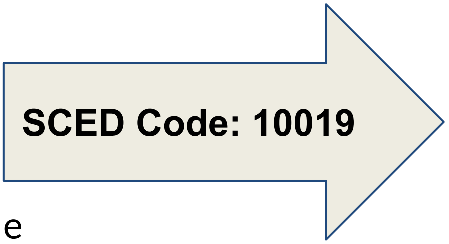
Group 2 Courses:

- May align to CSTA Standards as well as other courses
- May have less direct focus on computer science

Classifying Courses - Example 1

AP Computer Science Principles

AP Computer Science Principles introduces students to the central ideas of CS, **fostering computational thinking and inviting students to understand how computing changes the world**. Students are encouraged to apply creative processes when developing **computational artifacts** and while using simulations to explore questions of interest. There is a focus on using technology and programming as a means to solve problems. This course highlights the relevance of computer science by emphasizing the vital impact advances in computing have on people and society. Students also have the opportunity to investigate the innovations in other fields that computing has made possible and examine the ethical implications of new computing technologies. Fundamental course components are: **creative thinking, abstraction, data, algorithms, programming, the internet and societal impact**.



SCED Code: 10019

10019: AP Computer Science Principles

Following the College Board's suggested curriculum designed to parallel college-level computer science principles courses, **AP Computer Science Principles** courses introduce students to the fundamental ideas of computer science and **how to apply computational thinking across multiple disciplines**. These courses teach students to apply creative designs and innovative solutions when developing **computational artifacts**. These courses cover such topics as **creative development, data, algorithms and programming, computer systems and networks, and the impact of computing**.

Classifying Courses - Example 2

Honors Python

This course is designed to develop structured programming skills in **Python**. Learning to program offers the student an invaluable opportunity to develop **problem solving skills**. The process of defining a problem, breaking it down into a series of smaller problems and finally writing a computer program to solve it, is an **exercise in learning to think logically**. Topics will include: Python program **structure**, using variables, controlling program flow (loops), functions and arrays.

SCED Code: 10156

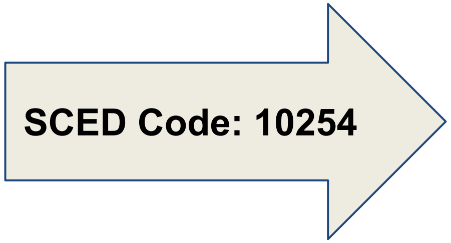
10156: Computer Programming - Other Language

Courses provide students with the opportunity to gain expertise in computer programs using languages other than those specified (such as Pascal, FORTRAN, **Python**, or emerging languages). As with other computer programming courses, the **emphasis is on how to structure** and document computer programs, using **problem-solving techniques**. As students advance, they learn how to best utilize the features and strengths of the language being used.

Classifying Courses - Example 3

Computer Technology - Hardware

In this hands-on course students **build and repair computers utilizing modern concepts in PC architecture**. Students will research, design, and implement computer and **networking systems**, learning the **latest PC hardware and software technologies**.



SCED Code: 10254

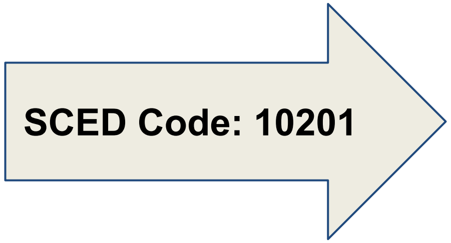
10254: IT Essentials: PC Hardware and Software

IT Essentials: **PC Hardware and Software courses provide students with in-depth exposure to computer hardware and operating systems**. Course topics include the functionality of hardware and software components as well as suggested best practices in maintenance and safety issues. Students learn to assemble and configure a computer, install operating systems and software, and **troubleshoot hardware and software problems**. In addition, these courses introduce students to **networking** and often prepare them for industry certification.

Classifying Courses - Example 4

Web Design

With the ever-growing influence of the Internet, students need to prepare themselves for a web-based future. Through this course, students learn how to **design**, create, and manipulate **web sites** for personal and professional applications. In addition to **basic HTML code**, this course will include site design and management using Adobe Dreamweaver.


SCED Code: 10201

10201: Web Page Design

Web Page Design courses teach students how to **design websites** by introducing them to and refining their knowledge of site planning, page layout, graphic design, and the use of markup languages—**such as Extensible Hypertext Markup, JavaScript, Dynamic HTML, Document Object Model, and Cascading Style Sheets**—to develop and maintain a web page. These courses may also cover security and privacy issues, copyright infringement, trademarks, and other legal issues relating to the use of the Internet. Advanced topics may include the use of forms and scripts for database access, transfer methods, and networking fundamentals.

Expanding Computing Education Pathways