

Getting Started Guide

Thank you for using CompuScholar's online Computer Science and Digital Literacy courses! This brief guide explains how to use the system and find related information.

Available Courses

For a list of available CompuScholar courses, please see our Course Overview page: <https://www.compuscholar.com/schools/courses/overview/>

Login Information and Online Delivery

All course material is delivered through an online Learning Management System (LMS). Our LMS can be accessed from any location with a web browser and Internet connection. Student or teacher logins are required to access course material.

Hardware and Software Requirements

The specific hardware and software requirements vary for each course. For more details, please see: https://learning.compuscholar.com/course_includes/Minimum_Requirements.pdf

3rd Party Software

All courses include hands-on computing activities or programming projects. Many of these activities can be completed online, directly within a web browser. If 3rd party software is required to complete any activity, the course will contain step-by-step download and installation instructions or online access links. All 3rd party software is **free** to use from any student or teacher computer.

Schools may choose to pre-install 3rd party software on common or shared school computers. Students are also encouraged to install the software at home in order to work outside the classroom.

Login Page

To access course material, please use an HTML5-compliant web browser such as Google Chrome, Microsoft Edge, Mozilla Firefox, or Apple Safari. Navigate to the CompuScholar login page provided with your course access instructions.

Your login page will look similar to the image below. Type your username and password into the “Log in” panel and click on the “Log in” button to continue.




Join a Course ▾ Help ▾ Main Website English (United States) (en_us) ▾

Latest News

Please click on each line item to read important announcements

Log in

 Username

 Password

Log in

[Forgotten your password?](#)

Cookies must be enabled in your browser

Main Dashboard

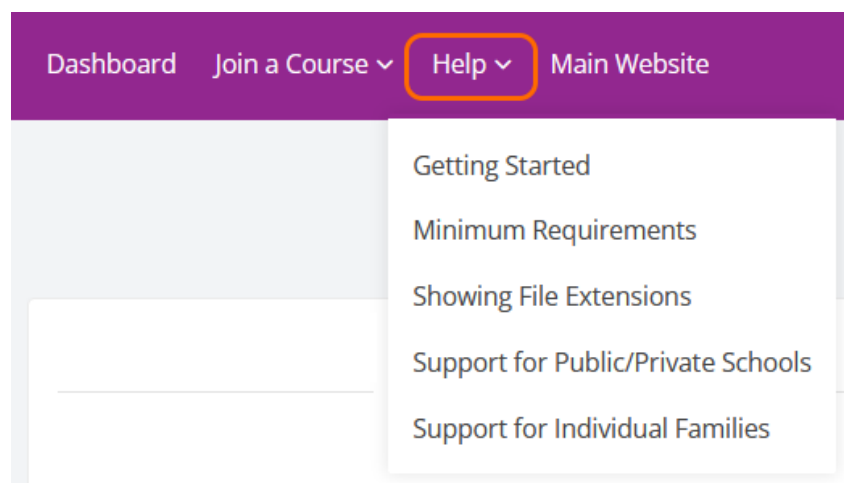
Once logged in, your Dashboard will display the courses that are assigned to your login.



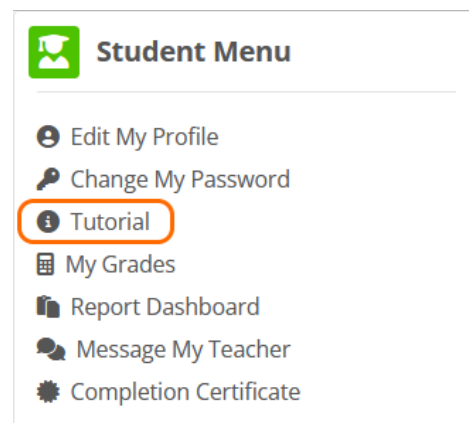
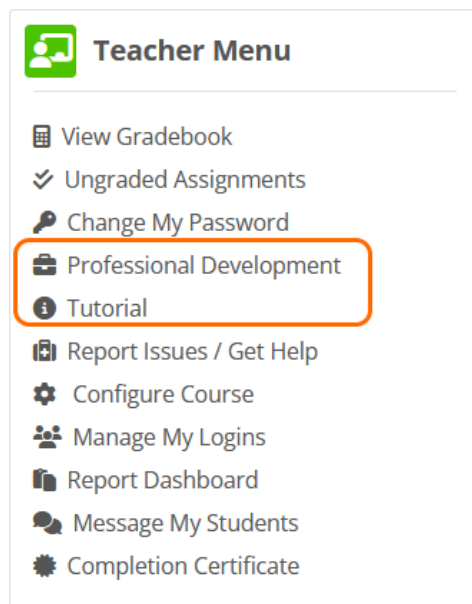
Simply click on the desired course (e.g. "Java Programming") to access that course material.

Online Help, Tutorials, and Professional Development

At the top of every course page is a standard toolbar with a **“Help”** selection. Inside **“Help”** are links to useful documents such as this guide, minimum hardware and software requirements, and our support portals.



Within each course, you will find a **Teacher Menu** or a **Student Menu** on the left. Both menus contain links to **“Tutorials”** or **“Professional Development”** modules that show students and teachers how to use the online system.



Getting Help

To get help, click on the appropriate “Support...” link from the top Help menu or visit our Support Portals directly.

<https://support.compuscholar.com/> (public/private schools)

<https://homeschoolhelp.compuscholar.com/> (homeschool)

Course Home Page

From a course home page, you will find one or more “**Course Planner and Syllabus**” documents pinned to the top. These documents contain detailed pacing guidance. In the main content area, you will see a list of individual chapters. Click on a chapter heading to access the lessons, activities, quizzes, and tests for that chapter.



AP Computer Science A - Course Planner and Syllabus (CB Endorsed)



Introductory Computer Science - Course Planner and Syllabus



Chapter One: Computing Concepts



Welcome to *Java Programming!* In this first chapter we will review common hardware and software terms and discuss computer ethics and security.



Chapter Two: Getting Started with Java



In this chapter, we will discuss common programming languages and the Java platform. You will write your first Java program and learn how to find online help documentation.


Chapter Page

Each chapter generally has 3 or more lessons arranged top-to-bottom. Each lesson typically has the following links:



- Lesson Video (optional viewing for students)
- Lesson Text (required reading)
- Quiz
- Teacher Guide
- Quiz Answer Key



Lesson One: Common Programming Languages

 Chapter 2, Lesson 1 Video

 Chapter 2, Lesson 1 Text

 Chapter 2, Lesson 1 Quiz  Hidden from students 

 Chapter 2, Lesson 1 Teacher Guide  Hidden from students

 Chapter 2, Lesson 1 Quiz Answer Key  Hidden from students

Lesson Videos (where present) and **Lesson Text** pages are student-facing material. **Quizzes** will either be visible initially or can be made visible by teachers at the appropriate time. **Teacher's Guides** and **Quiz Answer Keys** are only visible to teachers.

Lesson Videos

Lesson videos are multi-media alternatives to the lesson text. They are **optional** viewing for students and do not introduce new concepts. Audio-visual students may find them to be a useful introduction to and re-enforcement of the lesson text.

The videos are in a standard MP4 format and should play in any HTML5-compliant web browser. When you click on a “Lesson Video” link, a pop-up or tab window will show the streaming video. The video controls are browser-dependent and typically located at the bottom of the screen. Simply close the window or tab when finished.



Instructional videos have both **English** and **Spanish** captions – use your standard browser video controls to select or disable captions.

Lesson Text

The “Lesson Text” links will launch the reading material for the lesson in HTML format. The Lesson Text is **required reading** and contains detailed explanations and examples for each concept. Simply scroll down to read the full text and close the pop-up window or tab when finished.

Chapter Two: Getting Started with Java

Lesson Three: Writing Your First Program

Programming languages are like spoken languages; they need to be **standardized** to be universally understood. What does that mean? To answer that, let's look at the standards for spoken language. For languages such as English to be effective, everyone needs to agree on a common set of rules for **spelling, grammar, and vocabulary**. Otherwise, English speakers in different parts of the world could not understand each other. Programming languages such as Java are no different. A common set of rules defines how to write Java source code and compile that source code into programs.

Source Files

Nearly all files on your hard drive have some sort of **file extension**. Native executable programs may end in ".exe" or ".app", images might use ".jpg" or ".gif", and audio files may come as ".wav" or ".mp3". Sometimes operating systems such as Windows or Mac OS will hide these file extensions from the end-user. This may be fine for normal users, but you are a programmer! Programmers care a great deal about the file extensions because each one means something important.

Source code is human-readable text that follows the rules of a programming language. Source code can be written in a simple text editor and saved in text files on disk. To identify the contents of these files, each language uses a special file extension to show what programming language is in the file. All Java source files end with the **".java"** extension. The Java compiler knows that all files saved with a ".java" extension contain Java source code.

A Simple Java Source File

Consider the following sample Java source file. If stored on your computer, you would name this file "HelloWorld.java".

```
1 public class HelloWorld
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello, World");
```

Interactive Lesson Elements

Some courses contain interactive coding opportunities directly within the lesson text. These live coding panels contain code that can be modified and run within the web browser (no software installation is required).

Try It Now

```
1 public class HelloWorld
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello, World");
6
7         // add your code here
8
9     }
10 }
```

Run Code

Console Output...

In-Lesson Exercises

Many **Lesson Text** pages contain built-in, **hands-on activities** for students. These exercises are easily identified in blue **“Work with Me”** panels. Depending on the course, the specific exercise may be completed directly in the web browser with interactive coding or may be completed outside the course using the student’s own computing resources.



Work with Me: Creating a Class with main()

We have demonstrated how to create a simple class with a **main()** method. The **main()** method can contain one or more program statements. In this exercise, you are going to practice creating your own class and **main()** method with several statements.

Take the following steps to complete this activity

- In the source panel below, begin by creating a multi-line comment with your name, today’s date, and any other information you’d like to add.



Lesson Quizzes

Lesson quizzes are generally 5-question multiple choice or similar questions designed to assess the student's understanding of the just-completed lesson. After clicking on "Lesson Quiz" you will be taken to a screen where you can launch the quiz with a button similar to the one shown to the right.

Attempt quiz

You will then be presented with the quiz questions:

What is the name of the method that will automatically run when the program starts?

- a. class
- b. start
- c. program
- d. main

Students can scroll down to the bottom and click the "Finish attempt..." button to see a summary of answers and submit final answers for grading. When all questions have been answered, click on "Submit all and finish" to complete the quiz.

Teachers can "complete" quizzes as a preview as well, but no grades are sent to the electronic grade book!

Teachers have access to a "**Quiz Answer Key**" link underneath the student-facing quiz. The answer key is a PDF showing all questions and correct answers.

The a/b/c/d order of each question's answers is randomized on every student attempt.

Teacher Guide

The “**Teacher Guide**” link is visible only to teachers and leads to an HTML page containing lesson objectives, suggested classroom discussion questions, and other guidance on conducting the lesson. If any solutions are needed for the in-lesson “Work with Me” exercises, those solutions will be found in the Teacher’s Guide as well.

Teacher Notes - Lesson Three: Writing Your First Program

Programming languages are like spoken languages; they need to be standardized to be universally understood. A common set of rules defines how to write Java source code and compile that source code into programs. In this lesson, we take a look at the common syntax and rules for Java programming.

Lesson Objectives

- Understand how to use proper programming style to enhance the readability and functionality of the code
- Explain the use of internal comments, white space, and indentation
- Understand the concept of a source file
- Introduce the concept of a Java class
- Explain the importance of a "main" function in a console program
- Learn to use the **System.out.println()** function to display simple lines of text

AP Computer Science A - Learning Objectives and Essential Knowledge

Learning Objective	Essential Knowledge
1.1.A Represent patterns and algorithms found in everyday life using written language or diagrams.	1.1.A.1 Algorithms define step-by-step processes to follow when completing a task or solving a problem. These algorithms can be represented using written language or diagrams.
	1.1.A.2 Sequencing defines an order for when steps in a process are completed. Steps in a process are completed one at a time.

Teaching

Class Discussion

In a class discussion, make sure your students understand the basic points of the lesson. They should be able to answer the following questions:

1. Why do we need rules for programming languages?
2. What is a source file?
3. What is a file extension? How are these extensions used by operating systems?
4. What is a class and how do you define one in code?
5. What is the purpose of a main function?
6. What program statement will display text to the output window?

Work With Me Exercise


In this exercise, students will create their very first Java program from scratch, including a class, main() method, and multiple output statements. A sample solution can be viewed by clicking the "Show Solution" button within the exercise.

Chapter Homework

Many courses contain “**Homework**” problems near the end of each lesson. Homework problems provide **highly recommended reinforcements and practice** of the lesson concepts. Click on the “Homework” link to view all problems for the current chapter.



Chapter Two Homework

 Chapter Two Homework



Students will be taken directly to the list of problems. Teachers will first see an intermediate page with a list of student submissions (if any) and will click on “**Teacher Preview**” at the top to reach the list of problems.

Chapter Two Homework - Java Programming-2023

[Teacher Preview](#)

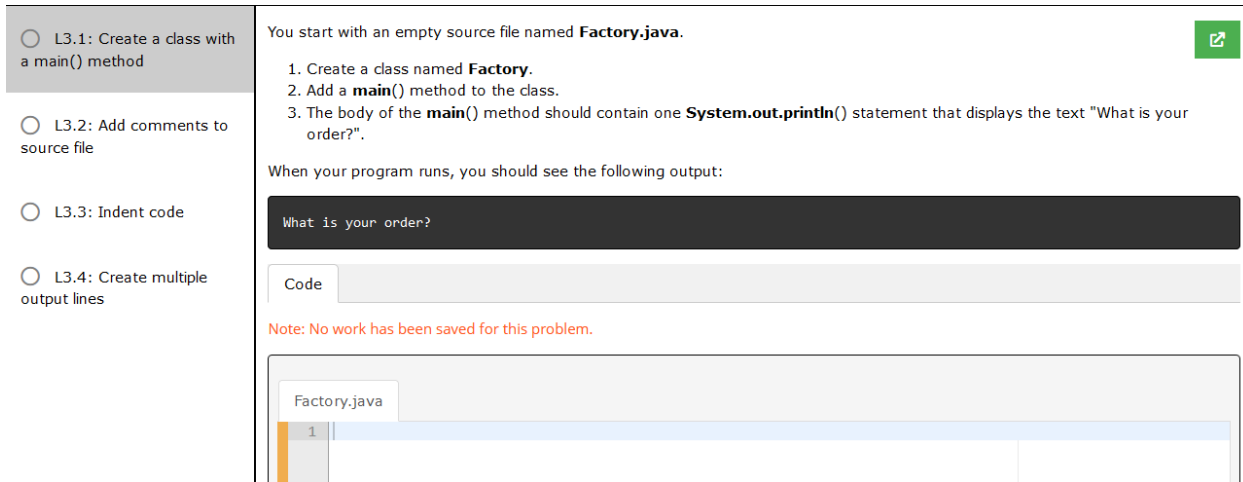
[View Solution](#)

[Configure Assignment](#)

[Refresh Student Info](#)

First name ▲▼	Last name ▲▼	Date Last Saved	Submission Status	Grade
Joe	Student	N/A	Completed 0 of 4	
Serious	Student	N/A	Completed 1 of 4	25

When viewing the homework problems, a tabbed list near the left will allow students or teachers to navigate between each problem.



Clicking on each tab will display the problem description to the right and an online interface for submitting answers. The main types of homework problems are:

- **Coding** (auto-graded by our system)
- **Drag-n-drop** (auto-graded by our system)
- **Free response Q&A** (teacher-graded)

Teachers have full control over homework grades. For details, please refer to the **“Professional Development”** modules linked from the **Teacher’s Menu** within each course.

Activities

Each chapter contains one or more **Activities** that provide students with hands-on opportunities to exercise their newly learned skills. Each Activity has an “**Activity Instructions**” page with detailed, student-facing instructions.

The link **underneath the Activity Instructions** allows students to access context-specific submissions.

- **Online activities** lead to an interface and problem options similar to **Homework** (see above).
- **External activities** lead to an interface where students can upload files for teacher review



Activity: Shopping List



Chapter 2 Activity Instructions



Shopping List Activity



Chapter 2 Activity Solution Guide **Hidden from students**

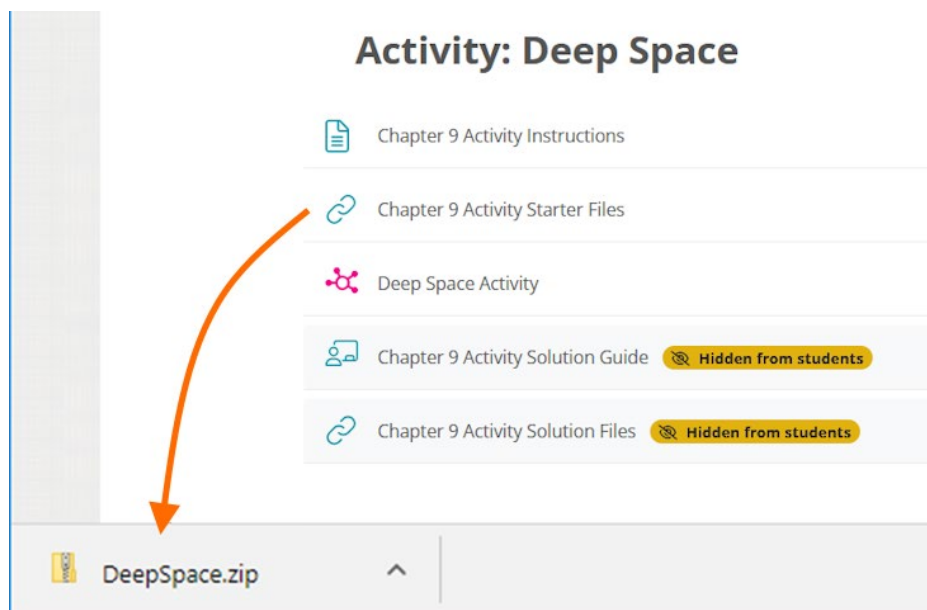


Each activity also has an “**Activity Solution Guide**” with teacher-facing guidance and answers for the activity. Some activities additionally have an “**Activity Solution Files**” link where teachers can download a ZIP demonstrating a solution to an externally completed activity.

Activity Starter Files and Solution Files

For **online** activities completed directly in the web browser, all “starter” files provided for students are accessed through the online coding environment. Similarly, online activity solution files for teachers are viewed online through the teacher’s activity menu.

For **external** activities completed outside our system (e.g. in a locally installed IDE), the course may provide a ZIP of starter files for the student or a ZIP of solution files for the teacher. When downloading ZIP files for external work, clicking on the “**Activity Files**” link will start the download process, the details of which will vary depending on your operating system and web browser.



The student ZIP file contains starting material the student will use locally to complete an activity or project. The student should save the file locally and un-zip it to their working area on their computer to get started on the activity. The teachers’ solution ZIP files contain a fully coded example of how students can meet the activity requirements.

Submitting Activities

For **online** activities completed directly in the web browser, students will submit their online work using the “**Submit**” button visible near the bottom of each problem.

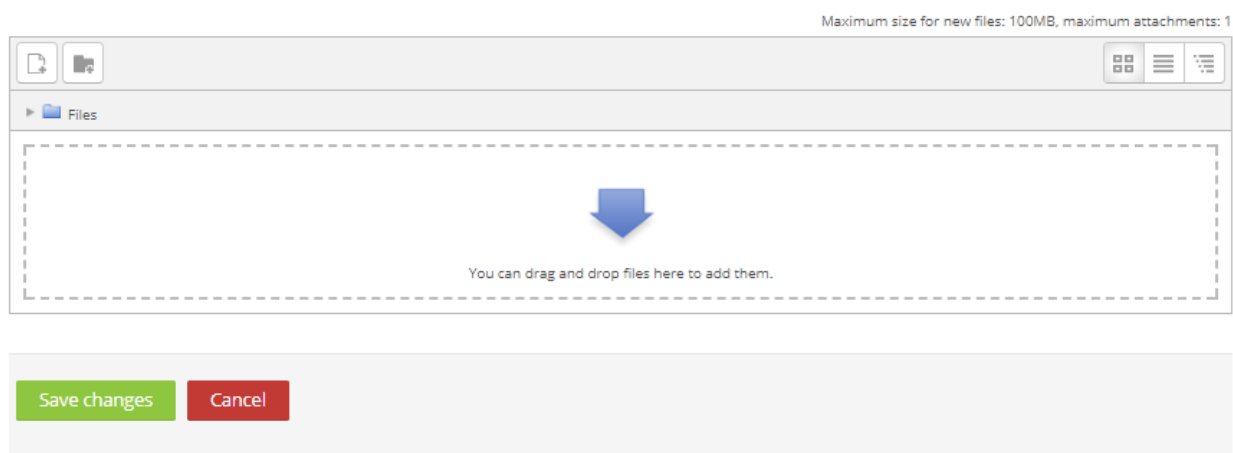


For **auto-graded** problems, the system will immediately assign and display a grade, along with a list of passed or failed test cases. For **teacher-graded** problems, a grade will not be visible until the teacher reviews and grades the student's work.

In other cases where work is done **externally**, student projects are typically bundled into a ZIP on the local computer and uploaded for teacher review. This can be accomplished by clicking on the activity link and then the “**Add Submission**” button. You will see an area where a file can be uploaded to the system.

Deep Space Activity

File submissions








For additional help on submitting or managing all kinds of activities, please see the “**Tutorial**” and “**Professional Development**” links from your Student or Teacher Menu.

Chapter Exams (Tests)

Each chapter typically ends with an Exam (test), similar in format to the lesson quizzes. The teacher is provided with an **Exam Answer Key**, which again has the same PDF question and answer-key format as the quizzes.

Chapter Two Exam

	Chapter 2 Exam	 Hidden from students	 <input type="checkbox"/>
	Chapter 2 Exam Answer Key	 Hidden from students	