



TeenCoder: Windows and Game Programming

Online Course Syllabus and Planner

Updated November, 2015

Online Course Overview

Course Title: *TeenCoder: Windows Programming, TeenCoder: Game Programming*

Online ISBNs: 978-0-9887070-0-9, 978-0-9887070-1-6, published 2015 by CompuScholar, Inc.

Length: 2 semesters

Student Pre-Requisites: Basic computer literacy skills, 9th – 12th grade status.

Description: The *TeenCoder: Windows Programming* and *TeenCoder: Game Programming* courses are a two-semester online bundle covering introductory C# programming and video game design topics.

Other introductory programming courses are not required; students merely need to have typical computer usage skills prior to starting the first-semester *TeenCoder: Windows Programming* class.

Materials:

- Online Student and Teacher logins
- Microsoft Visual Studio (free IDE), XNA Game Studio
- Windows personal computer
- Course activities (hands-on programming assignments)
- Course instructional videos, lesson text, and supplemental documentation

Labs and Grading

Every chapter contains one or more hands-on programming labs where students will create programs to demonstrate understanding of the lesson topics. Projects begin simply with C# applications and expand into video games in the second semester. These labs, combined with individual lesson quizzes and end-of-chapter tests, are used to determine the student grade.



Homeschool Programming, Inc.

Please Note: Some course lessons and activities may include opportunities for students to work in teams or small groups. If your student is working individually (e.g. as a homeschooler), simply complete those activities as an individual. One student can perform each of the tasks that would be spread across multiple team members in a group setting. Other lesson sections with peer-supported vocabulary re-enforcement or feedback tasks can be skipped or completed with a teacher, if desired.

Course Planner

A typical school semester consists of approximately 36 calendar weeks or 180 days of school. This course plan covers a full 35+ weeks with core material, assuming students are working 3-5 hours per week to stay on pace. Some students may move faster or slower than the suggested pace.

Teachers may choose to add Supplemental Lessons as desired to meet student interests.

TeenCoder: Windows Programming – First Semester

Week	Reading and Objectives	Labs
1	<p>Chapter One: Understanding Your Computer</p> <ul style="list-style-type: none">• Computers Past and Present• Survey of Computer Hardware• Introduction to Computer software• Common Programming Languages• Computer Ethics and Security	<p>Establish Development Environment - Install Microsoft Visual C# Express, create working directory, practice submitting projects through the online interface.</p> <p>Class discussion and review of a sample EULA terms and conditions.</p>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
2	<p>Chapter Two: Fundamentals of C#</p> <ul style="list-style-type: none">• Introduction to C#• Your Integrated Development Environment• Console Programs• Getting Console Input	<p>Echo, Echo – The student’s first C# program will use the console to ask the user for a name and print a message as output.</p>
3	<p>Chapter Three: Windows Programming Concepts</p> <ul style="list-style-type: none">• Your First GUI Program• Common Windows Elements• Event-Driven Programming• Namespaces	<p>A More Personal Hello – The student will create an even-driven GUI program to display their name in a message box in response to a button click.</p>
4	<p>Chapter Four: Data Types and Variables</p> <ul style="list-style-type: none">• Value Data Types• Variables• Reference Data Types• Introducing Strings• The Binary Number System	<p>Experiment with Data Types – The student will demonstrate declaring, initializing, and printing variables of different data types.</p>
5	<p>Chapter Five: Basic Flow Control</p> <ul style="list-style-type: none">• Logical Expressions• Using the “if” Statement• For Loops• While Loops	<p>Jeepers, Beepers – The student will create loops of different types to produce a specific number of beeps or pop-ups entered by the user.</p>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
6	<p>Chapter Six: User Input</p> <ul style="list-style-type: none">• Text Boxes• List Boxes and Combo Boxes• Radio Buttons and Check Boxes	<p>Telling Tall Tales – The student will create a “mad-lib” style program to demonstrate a variety of user input controls.</p>
7	<p>Chapter Seven: Math Functions in C#</p> <ul style="list-style-type: none">• Math Operators (+, -, *, /, and %)• .NET Framework Math Functions• Common Algorithms• A Simple Calculator	<p>Algorithms Practice – The student will flowchart and implement two simple algorithms.</p> <p>Divide and Multiply – The student will add multiplication and division buttons to the calculator created in the last lesson.</p>
8	<p>Chapter Eight: Working with Strings</p> <ul style="list-style-type: none">• Common String Operations• Formatting Strings• Converting Between Strings and Numbers	<p>Caesar’s Cipher – The student will write a program that translates a message to a cipher code and then back to plain form again.</p>
9	<p>Chapter Nine: Methods</p> <ul style="list-style-type: none">• Writing and Calling Methods• Method Parameters and Return Values• Calling Methods	<p>What’s Your Birthday? – The student will write a function to translate a date into a day of the week.</p>
10	<p>Chapter Ten: Debugging and Exceptions</p> <ul style="list-style-type: none">• The Visual C# 2010 Express Debugger• Debugging Demonstration• C# Runtime Exceptions• Finding Runtime Errors	<p>Divide by Zero – In this lab the student will identify and resolve a hidden error left in an earlier program.</p>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
11	Chapter Eleven: Collections <ul style="list-style-type: none">• Arrays• Linked Lists• Enumerations and ForEach	Your ToDo List – The student will create a program that allows users to add and removed text items from a linked list.
12	Chapter Twelve: Object-Oriented Programming <ul style="list-style-type: none">• Object-Oriented Concepts• History of OOP• Designing an Object	Creating Songs – The student's will design on paper Song and Note classes and assemble Songs from multiple Notes.
13-14	Chapter Thirteen: Classes in C# <ul style="list-style-type: none">• Defining a Class• Properties and Methods• Public vs. Private• Constructors• Static Members	Your Song Player – The student will implement a Song class to play music defined in the previous chapter.
15	Chapter Fourteen: Sorting and Recursion <ul style="list-style-type: none">• Simple Sorting• Recursion• Recursive Sorting	The Number Sort – The student will write an Insertion Sort function to sort numbers in a list.



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
16	<p>Chapter Fifteen: Vector Graphics</p> <ul style="list-style-type: none">• Screen Coordinates• Drawing Lines• Drawing Circles and Rectangles	<p>Graphing – The student will create simple line graphs from input data sets.</p>
17	<p>Chapter Sixteen: Inheritance and Polymorphism</p> <ul style="list-style-type: none">• Base Classes and Derived Classes• Using References to Base and Derived Classes• Virtual Base Methods• The “Object” Base Class• Using Base Features from Derived Classes	<p>Creating the Chess Pieces – The student will create a small object hierarchy of standard chess pieces in preparation for the final project.</p>
18	<p>Chapter Seventeen: Final Project</p> <p>For the final project the student will complete a Chess game. The student will create the abstract hierarchy of pieces (AbstractChessPiece, Pawn, Knight, Rook, etc.) and write other logic to complete the game.</p> <p>The project consists of 5 guided lab steps. Each guided step contains a checkpoint for testing to ensure code meets the requirements at each step.</p>	<p>Starting Your Chess Project – Ensure the student can build the starter project.</p> <p>Initializing the Game() – Write logic to initialize the game and set pieces in the starting position.</p> <p>Finishing handleClick() – Write game logic to allow selection and de-selection of game pieces.</p> <p>Moving Pawns – Write game logic to control Pawn movement.</p> <p>Moving Other Pieces and Testing for Check – Write game logic to control movement and capture of other pieces. Also complete logic to test for check.</p>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
	<p>Supplemental Lessons and Projects</p> <p>The supplemental lessons provide additional opportunities for students to work on group or individual projects with real-world data. They are recommended after Chapter 15, and can be scheduled at teacher convenience or given as additional homework.</p> <ul style="list-style-type: none">• Tracking and Plotting Temperature Changes• Stock Market Tracking and Prediction• Software Development Careers• Technical Writing• File I/O• Software Development Process	<p>Temperature Graph – The student will plot real historical temperatures for a location over a time range.</p> <p>Stock Market Trading – The student will write a game enabling others to make daily trades against historical stock prices.</p> <p>Career Exploration Activity – The student will research software development careers.</p> <p>Technical Writing Exercise – Practice writing a technical paper.</p> <p>Song Database – Create a database of songs using a CSV file.</p> <p>Your SDLC Docs – Practice writing documents found during a normal software development lifecycle.</p>

After students complete the first-semester **Windows Programming** material, they can move to the second-semester **Game Programming** course.



Homeschool Programming, Inc.

TeenCoder: Game Programming – Second Semester

Week	Reading and Objectives	Labs
1	<p>Chapter One: Introduction to Game Programming</p> <ul style="list-style-type: none">• What You Already Know• Types of Computer Games• What You Will Learn in this Course• Introduction to XNA• Computer Ethics and Security	<p>Install XNA Game Studio - Install Microsoft Visual C# Express, XNA Game Studio, and create a working directory.</p> <p>Class discussion and review of a sample EULA terms and conditions.</p>
2	<p>Chapter Two: Game Design</p> <ul style="list-style-type: none">• The Game Proposal• The Game Engine• Creating an XNA Game Project• The Game Loop	<p>Your Game Design – The student will brainstorm and design key game sequences on paper.</p> <p>Looping Colors – The student use a game loop to periodically change colors on the screen.</p>
3	<p>Chapter Three: Graphics Concepts</p> <ul style="list-style-type: none">• Screen Coordinates• Colored Pixels• Drawing Circles, Lines, and Polygons• Video Displays	<p>Vector Tree – The student will create a unique tree using vector graphics and recursion.</p>
4	<p>Chapter Four: Working with Images</p> <ul style="list-style-type: none">• Surfing the Content Pipeline• Drawing Images• Image Transformations• Drawing Text	<p>Starry Night – The student will demonstrate scaling and rotating pixels and displaying transparent graphics to create a landscape.</p>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
5	<p>Chapter Five: User Input</p> <ul style="list-style-type: none">• Keyboard Input• Mouse Input• XBox 360 Controller	<p>Cat and Mouse – The student will create a simple chase game to demonstrate keyboard and mouse processing.</p>
6	<p>Chapter Six: Sprites</p> <ul style="list-style-type: none">• Introducing Sprites• The Swarm Game• Initializing Your Swarm• Sprite Movement	<p>Raising the Swarm – The student will write code to initialize the game screen and main objects.</p> <p>Buzzing Bees – The student will write code to make the bees move on the screen.</p>
7-8	<p>Chapter Seven: Movement, Firing, and Collisions</p> <ul style="list-style-type: none">• Adding Player Control• Shooting Stingers and Smoke• Collision Detection	<p>Sliding Smoke Gun – The student make the smoke gun move in response to keyboard commands.</p> <p>Shooting the Swarm – The student will make the smoke gun fire shots in response to keyboard commands.</p> <p>Feeling the Sting – The student will add collision detection between sprites in Swarm.</p>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
9	<p>Chapter Eight: Player Progression</p> <ul style="list-style-type: none">• Ending and Re-Starting the Game• High Scores• Progressive Difficulty	<p>Ending the Game – The student will write end-of-game logic for Swarm.</p> <p>Tracking Swarm High Scores – The student will write code to track high scores in Swarm.</p> <p>Challenge Levels – The student will adjust the game difficulty based on the player winning or losing the previous level.</p>
10	<p>Chapter Nine: Animation</p> <ul style="list-style-type: none">• Animation Concepts• Animation Textures• Animation in the Sprite Class	<p>Animating the Swarm – The student will add animated images to several sprites in the Swarm game.</p>
11	<p>Chapter Ten: Music and Sound Effects</p> <ul style="list-style-type: none">• Sound Files• Playing Sound Effects• Playing Music• The XACT Tool	<p>Audible Swarm – The student will add background music and sound effects to the Swarm game.</p>
12	<p>Chapter Eleven: Game Physics</p> <ul style="list-style-type: none">• Velocity and Acceleration• Gravity and Wind• Reflection	<p>Snowball Fight – The student will implement logic for a classic artillery-style game.</p>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
13	<p>Chapter Twelve: Maze Generation</p> <ul style="list-style-type: none">• Maze Types• Generating a Perfect Maze• Solving a Perfect Maze	<p>A-Maze-Ing Backtracker – The student's will write recursive backtracker algorithms to generate and solve a maze.</p>
14	<p>Chapter Thirteen: Menus, Overlays, and Deployment</p> <ul style="list-style-type: none">• Title Screens and Option Menus• Handling Different Screens• Displaying Scores and Overlays• Distributing Games• Game Testing	<p>Tic-Tac-Toe – The student will implement a tic-tac-toe game that demonstrates title screens, menus, scoring, and overlays.</p>
15	<p>Chapter Fourteen: Multiplayer Games</p> <ul style="list-style-type: none">• Handling Multiple Inputs• Scrolling Games• Viewports and Cameras	<p>Star Racer – The student will write the camera and viewport logic for a vertically scrolling racing game.</p>
16	<p>Chapter Fifteen: Artificial Intelligence</p> <ul style="list-style-type: none">• Understanding AI• Developing an AI Algorithm• Simple Movement Algorithms• AI for Star Racer	<p>Star Racer AI – The student will improve the existing Star Racer AI routines from the previous lesson.</p>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
17	<p>Chapter Sixteen: Final Project</p> <p>For a final project the student will create an arcade game called “Bumper Cars”. The project contains a series of guided steps as activities. The student can test their program at each step to ensure each feature works correctly.</p> <ul style="list-style-type: none">• Bumper Cars Overview• Menus and Controls• Adding Cars• Oil Slicks and Coins• Bumper Car Sound Effects• Add Artificial Intelligence	<p>Project Kick-Off – Build and run the activity starter project.</p> <p>What’s on the Menu? – The student will implement the game menu system.</p> <p>Start Your Engines – The student will add the player car objects to the game.</p> <p>Hazards and Rewards – The student will add the coins as targets and oil slicks as hazards.</p> <p>Make Some Noise – The student will add background music and sound effects.</p> <p>Racing Buddy – The student will write an AI routine to give the second car an auto-pilot option.</p>
	<p>Supplemental Lessons and Projects</p> <p>The supplemental lessons provide additional opportunities for students to work on group or individual projects to research existing games or role-play in a simulated game company.</p> <p>Lessons and projects can be scheduled at teacher convenience (even early in the semester); no code will be written.</p> <ul style="list-style-type: none">• Running a Game Business• Platforms, Tools, and Networking• ESRB Rating System• Computer Game History	<p>Game Company Study – Students will simulate research and development activity in a game company using a variety of team roles.</p> <p>Online Game Evaluation – The student will evaluate online games using a variety of criteria.</p>