



TeenCoder: Windows Programming

Printed Course Syllabus and Planner

Updated October, 2015

Printed Course Overview

Course Title: *TeenCoder: Windows Programming*

Textbook ISBN: 978-0-9887033-1-5, published 2013 by Homeschool Programming, Inc.

Length: 1 Semester

Student Pre-Requisites: Basic computer literacy skills, 9th – 12th grade status. Other introductory programming courses are not required.

Description: *TeenCoder: Windows Programming* is a one-semester course introducing computer programming skills and computer science topics using the C# language. This class is a pre-requisite for the follow-on *TeenCoder: Game Programming* class.

Materials:

- Student Textbook
- Visual Studio IDE
- Windows personal computer
- Course activities (hands-on programming assignments)
- Course supplemental documentation
- Course instructional videos (optional)

Labs and Grading

Every chapter contains one or more hands-on programming labs where students will design or implement programs to demonstrate understanding of the lesson topics. These labs, combined with end-of-chapter tests, are used to determine the student grade.



Homeschool Programming, Inc.

Course Planner

A typical school semester consists of approximately 18 calendar weeks or 90 days of school. This course plan covers the full 18 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.

Students may choose to implement a relaxed two-semester course by devoting less time per week or moving at a slower pace. The “Week” column in the planner below reflects a one-semester plan, and those on a two-semester plan can simply scale accordingly.

Week	Reading and Objectives	Labs
1	Chapter One: Introduction to Windows Programming <ul style="list-style-type: none">• History of Windows• The Evolution of Windows Programming• Windows Programming Languages	Install Development Environment Students will install the Microsoft Visual C# IDE (free download)
2	Chapter Two: Fundamentals of C# <ul style="list-style-type: none">• Introduction to C#• Visual C# Software• Hello World	Enhance Hello World The students will make changes to the Hello World application created during the chapter.
3	Chapter Three: Windows Programming Concepts <ul style="list-style-type: none">• Common Windows Elements• C# Syntax• Event-Driven Programming• Namespaces	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.



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
4	<p>Chapter Four: Data Types and Variables</p> <ul style="list-style-type: none">• Value Data Types• Variables• Reference Data Types• Introducing Strings	<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>
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• A Simple Calculator	<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>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
9	Chapter Nine: Methods <ul style="list-style-type: none">• Writing and Calling Methods• Method Parameters and Return Values• Calling Methods	What's Your Birthday? – The student will write a function to translate a date into a day of the week.
10	Chapter Ten: Debugging and Exceptions <ul style="list-style-type: none">• The Visual C# Debugger• Debugging Demonstration• C# Runtime Exceptions	Divide by Zero – In this lab the student will identify and resolve a hidden error left in an earlier program.
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.



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
15	<p>Chapter Fourteen: Sorting and Recursion</p> <ul style="list-style-type: none">• Simple Sorting• Recursion• Recursive Sorting	<p>The Number Sort – The student will write an Insertion Sort function to sort numbers in a list.</p>
16	<p>Chapter Fifteen: File Input and Output</p> <ul style="list-style-type: none">• Using Files in a Program• Reading and Writing Text Files• Reading and Writing Binary Files• SaveFileDialog and OpenFileDialog	<p>Your Own Notepad – The student will create a simple text editor that can load and save text files.</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>



Homeschool Programming, Inc.

Week	Reading and Objectives	Labs
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>