



Ephrata High School  
Course Syllabus  
Program Design  
#3105



### *I. Course Description*

The purpose of this course is to introduce students to the systematic design of computer programs utilizing the interactive and graphical programming environment of **DrScheme**. Students will learn to formulate goal statements and illustrate them with concrete examples; organize available data in the form of program templates; and then program and test their programs for mistakes. This approach to programming not only teaches students the fundamentals of program design but also strengthens their mathematical problem solving skills. **This course is an elective – no math credit. Completion of Intro – Visual Basic is recommend but not required.**

### *II. Materials & Equipment*

DrScheme Programming Software (free from pltscheme.org)

Access to a computer (either at home, or in the computer lab at school during 9<sup>th</sup> period) is recommended.

### *III. Course Goals & Objectives*

This main focus of this course is *the design process* that leads from problem statements to well-organized solutions; it does not emphasize the study of programming language details, algorithmic minutiae, and specific application domains. Instead, the objective is to focus on the design process. Students learn to move from a problem statement to a computational solution in step-by-step fashion with well-defined intermediate products. In the process they learn to read, to analyze, to organize, to experiment, to think in a systematic manner. In this course, a programming environment specifically developed for this style of introductory programming is used. It also grows with the students as they master more and more of the material.

### *IV. Course Topics (Summary Outline)*

#### I. Numbers, Expressions, Simple Programs

- Numbers and Arithmetic
- Variables and Programs
- Word Problems
- Errors
- Designing Programs

#### II. Programs are Function Plus Variable Definitions

- Composing Functions
- Variable Definitions

#### III. Conditional Expressions and Functions

- Booleans and Relations
- Functions that Test Conditions
- Conditionals and Conditional Functions
- Designing Conditional Functions

#### IV. Compound Data, Part 1: Structures

- Structure Definitions
- Data Definitions
- Designing Functions for Compound Data

#### V. The Varieties of Data

- Mixing and Distinguishing Data
- Designing Functions for Mixed Data
- Input Errors

#### VI. Compound Data: Lists

- Lists
- Data Definitions for Lists of Arbitrary Length
- Processing Lists of Arbitrary Length
- Designing Functions for Self-Referential Data Definitions
- Functions that Produce Lists

- Lists that Contain Structures

#### VII. Natural Numbers

- Defining Natural Numbers
- Processing Natural Numbers of Arbitrary Size

#### VIII. Composing Functions, Revisited Again

- Designing Complex Programs
- Generalizing Problems, Generalizing Functions

### *V. Assignments & Grading*

Scores will be accumulated throughout the year by quizzes, tests and submitted assignments and programs. Many of the programming assignments will be completed with a partner using Pair-Programming. Pair-Programming requires a willingness to work with others. Additionally, it develops good communication skills.