

Introduction to Programming

Course Text

There is no text for this course. All materials are included in the course fee.

Course Description

This course introduces programming concepts to students using the language of their choice: C++, Python, or Java while following a consistent course sequence, structure, and schedule of assessments.

The course teaches the core computer science concepts of variables, branching, loops, arrays/lists, and functions/methods. It also introduces object-oriented programming with classes and inheritance. The course covers use pointers and streams and teaches a variety of good coding practices, including iterative development, code formatting, and variable naming schemes.

Course Objectives

After completing this course, students will be able to:

- Explain the basics of how a computer works, then write a first program
- Use standard input and output, and understand common syntax errors
- Declare and initialize variables with valid identifiers
- Use a variety of integer data types and the concept of overflow. Use floating-point, character, and string data types
- Cast between data types
- Develop programs that branch based on user input
- Write expressions with relational and equality operators and develop expressions with logical operators
- Use multiple branches for more complex programs
- Use the Boolean data type to store results of conditional statements
- Access elements in arrays/lists
- Understand memory usage for single and multi-dimensional arrays/lists
- Combine loops and arrays/lists, and develop programs with multiple arrays/lists
- Write a first function/method, then return from a function/method and parameterize a function
- Learn reasons to use a function/method

- Combine functions/methods with branches and loops and use incremental development with functions/methods
- Send and use arguments passed to program
- Write classes that group functions/methods and data
- Initialize class variables with class constructor
- Introduce abstract data types
- Creating, copying, and deleting objects
- Understand standard input and output implementation
- Manipulate text and floating-points during output
- Read input from a string
- Input and output to a file

Course Prerequisites

StraighterLine suggests, though does not require, that students take Pre-Calculus or its equivalent before enrolling in this course.

Important Terms

In this course, different terms are used to designate tasks:

- **Reading:** The Participation Activities in the textbook, which are interactive activities and form the core learning material. Student graded on completion of Participation Activities.
- **Homework:** The Challenge Activities in the textbook, which are small coding problems designed to bridge the gap between reading and labs. Each problem is auto-graded.
- **Labs:** A set of graded programming tasks. Students write a program to perform a defined task. Each lab targets a particular concept. All labs are graded automatically.
- **Midterm:** A graded online test. Fully-automated grading.
- **Final:** A graded and proctored online test. Fully-automated grading.

Course Evaluation Criteria

StraighterLine does not apply letter grades. Students earn a percentage score. A passing percentage is **70%** or higher.

If you have chosen a Partner College to award credit for this course, your final grade will be based upon that college's grading scale. Only passing scores will be considered by Partner Colleges for an award of credit.

There are a **total of 1000 points** in the course:

Topic	Assessment	Points Available
1	Reading	10
1	Homework	15
1	Labs: <ul style="list-style-type: none">• Basics• Interleaved input / output• Formatted output: Hello World!• Formatted output: No parking sign	40
2	Reading	10
2	Homework	15
2	Labs: <ul style="list-style-type: none">• Divide by x• Driving costs• Expression for calories burned during workout• Using math functions/methods• Phone number breakdown• Simple statistics• Musical note frequencies	40
3	Reading	10
3	Homework	15
3	Labs: <ul style="list-style-type: none">• Remove gray from RGB• Largest number• Interstate highway numbers• Seasons• Exact change• Leap Year	40

4	Reading	10
4	Homework	15
4	Labs: <ul style="list-style-type: none">• Convert to binary• Mad Lib - loops• Password modifier• Count input length without spaces, periods, or commas• Countdown until matching digits• Output range with increment of 10• Print string in reverse	40
4	Midterm	80
5	Reading	10
5	Homework	15
5	Labs: <ul style="list-style-type: none">• Middle item• Output values below an amount• Adjust list by normalizing• Word frequencies• Elements in a range	40
6	Reading	10
6	Homework	15
6	Labs: <ul style="list-style-type: none">• Miles to track laps• A jiffy• Driving cost - functions• Swapping variables• Max magnitude• Convert to binary - functions	40
7	Reading	10

7	Homework	15
7	Labs: <ul style="list-style-type: none">• Triangle area comparison - classes• Car value - classes• Winning team - classes	40
8	Reading	10
8	Homework	15
8	Labs: <ul style="list-style-type: none">• Parsing strings• Data visualization	40
9	Final	400
Total		1000

Course Topics and Objectives*

Topic Number	Topic Title	Objectives
1	Introduction	<ul style="list-style-type: none">• Write first program• Use standard input and output• Interpret common syntax errors• Learn basics of how a computer works
2	Variables / Assignments	<ul style="list-style-type: none">• Declare variables with valid identifiers• Learn a variety of integer data types and concept of overflow Use floating-point, character, and string data types• Cast between data types

3	Branching	<ul style="list-style-type: none">• Develop programs that branches based on user input• Write expressions with relational and equality operators• Multiple branching for more complex programs• Develop expressions with logical operators• Use Boolean data type to store result of conditional statements
4	Loops	<ul style="list-style-type: none">• Use while loop to repeat code execution• Count using a while loop• Strengths of for-loops• Write nested loops• Use flow control to break or continue loops
5	Arrays/Lists	<ul style="list-style-type: none">• Access elements in arrays/lists• Combine loops and arrays /lists• Develop program with multiple arrays/lists• Understand memory usage for multi-dimensional arrays/lists
6	User-Defined Functions/Methods	<ul style="list-style-type: none">• Write first function/method• Return from a function/method and parameterize a function• Learn reasons to use a function/method• Combine functions/methods with branches and loops• Use incremental development with functions/methods• Send and use arguments passed to program
7	Classes	<ul style="list-style-type: none">• Write classes that group functions/methods and data• Initialize class variables with class constructors• Overload class constructor Introduce abstract data types

8	Streams	<ul style="list-style-type: none">• Understand standard input and output implementation• Manipulate text and floating-points during output• Read input from a string• Input and output to a file
9	Other topics as assigned	<ul style="list-style-type: none">• Final Exam

*-Course topic titles may vary slightly. Please see your course for more specific topics.