

TCSS 142 Master Syllabus
Version: February 2018
(Approved: 25 May 2018)

Course Title

Introduction to Programming

Catalog Description

Introduces the design and implementation of computer programs. Includes an introduction to program structure and data types. Prerequisite: minimum of 2.0 in TQS 120 or MATH 120, 68% on MPT-A test, 75% on MATHEC placement test, or a score of 2 on the AP exam.

Preconditions

- Analyze basic arithmetic and geometry problems.
- Correctly employ basic arithmetic and geometry knowledge to solve practical problems.
- Analyze basic algebra problems.
- Correctly employ basic algebra knowledge to solve practical problems.

Student Learning Goals (to be added to syllabus handed out to students)

- Write simple algorithms using pseudocode and flowcharts.
- Identify correct syntax and semantics of a high-level language.
- Analyze and explain the behavior of simple programs involving the fundamental programming constructs (variables, types, expressions, assignment, simple I/O, conditional and iterative control structures, functions and parameter passing, structured decomposition).
- Develop and implement programs involving the fundamental programming constructs (variables, types, expressions, assignment, simple I/O, conditional and iterative control structures, functions and parameter passing, structured decomposition).
- Analyze and explain the behavior of programs that use arrays, strings, and objects.
- Develop and implement programs that use arrays, strings, and objects.
- Debug and test simple programs

CSS Degree Student Learning Outcomes that this course contributes to (to be added to syllabus handed out to students)

- a. an ability to apply knowledge of computing and mathematics appropriate to the discipline;
- b. an ability to analyze a problem, identify and define the computing requirements appropriate to its solution;
- c. an ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs

UWT Student Learning Goals that this course contributes to (to be added to syllabus handed out to students)

Inquiry and Critical Thinking

Students will acquire skills and familiarity with modes of inquiry and examination from diverse disciplinary perspectives, enabling them to access, interpret, analyze, quantitatively reason, and synthesize information critically.

Topics covered

- Basic syntax and semantics of a high-level language
- Variables, types, expressions, and assignment
- Basics of data representation (integers, floats, characters, strings)
- Procedural decomposition of problems
- Defining methods: parameters, return values
- Conditional and iterative control structures
- Input/output: console, file
- Arrays (one-dimensional)
- Algorithmic thinking and pseudocode (including accumulator pattern, finding mean, median, and range)
- Use of standard libraries
- Debugging and testing
- Use of meaningful identifiers
- Time permitting: introduction to user defined abstract data types (e.g. classes)

Additional Information

The course is to be taught in a procedural language with a light syntax such as Python. This course has an associated lab section that meets once per week in addition to the 2 lectures per week.

Some of the possible textbooks:

An Introduction to Programming Using Python, David Schneider, Pearson, ISBN-10: 0134058224
Starting out with Python, Tony Gaddis, 4th ed, Pearson, ISBN-10: 0134444329
Building Java Programs 4th ed., Stuart Reges and Marty Stepp, Addison Wesley, ISBN-13: 9780134322766

Sample course schedule

Wk	Topic	Reading	Labs
1	Intro, basic types and variables, simple expressions		Lab 1
2	Interactive input, truth tables, selection (sequential and nested)		Lab 2
4	Strings and files – for each loops for strings and files and intro to nested for each loops Counting, accumulator pattern		Lab 4
5	Lists (basic ops) and files – for each loops and nested for each loops finding the min/max, summing up a list, mean, median, parallel lists, lists for tallying		Lab 5
6	Mid-term Exam (up to an including strings) Functions: void, local variables, parameter passing		Lab 6
7	Value-returning functions: libraries and programmer-defined, design by contract – pre and postconditions String idioms, returning booleans		Lab 7
8	Loops – while and for (basics ops and nested) Validating user input, count control, sentinels, flags, in-range, sequences, combining with function calls		Lab 8
9	More loops, intro to ADTs Loop patterns for strings and lists – combining lists, nested fors		Lab 9
10	Intro to Classes, Data Abstraction and Object-Oriented Programming Instance and class fields, passing objects as arguments, basic composition, mutable and immutable types		Lab 10
11	Finals week Final exam (Comprehensive)		