

CSC 120

Introduction to Programming with Java



Lecture 1

University of Mount Union
Math/Computer Science Department



Computer Science \neq Coding

Many people think that all computer careers involve the stereotype of the lone programmer sitting in front of a monitor for 12 hours a day, getting no sunlight, etc.

This is simply not true in today's world

Computing touches almost all aspects of modern life

There are many tech and tech-related careers where the majority of the work is something other than programming

Network Administration, Computer Security, Database Orchestration, Analytics, etc

These are high-paying careers with lots of growth potential for people with skills and knowledge



So why is our first course at Mount Union a Coding course?

Even though many careers in the tech sphere are not programming only jobs, knowledge of how programming works is critical to understanding the functioning of computers and computer applications

Why the structure of a network must be organized the way it is, or the reasons why a specific password policy is not secure, all are based on the programming of the software used in the system.

In order to work effectively as a computer professional in the tech sector, one needs a basic understanding of programming



CSC 120 is the foundation of most of the majors & minors in our Program

Basic Programming Skills are essential for success in our computing major and minor programs

CSC 120 is required for the Computer Science major and the MPSP major, as well as most of our minors

Several departments across the campus require this course for one or more of their programs:

Mathematics

Political Science National Security track

Computer Engineering

Electrical Engineering



Other Introductory Computing Courses at Mount Union (slide 1 of 2)

CSC 112: Introduction to Computer Science

Designed as a course to introduce various aspects of the field to students who are not intending to be a CS major

CSC 112 counts for the National Security major requirements, and is MUCH more appropriate for it

If you are here today because you are a National Security major, or you only are looking to earn a Computer and Network Security minor, you should think about taking CSC 112 instead.

>>> CSC 112 will be offered in Spring 2025



Other Introductory Computing Courses at Mount Union (slide 2 of 2)

CSC 108: Introduction to Computer Programming

A 2-credit-hour course for students who have absolutely no prior programming experience, and are not strong in mathematics or science subjects

A half-semester course that can act as a preparation for CSC 120 for students who would benefit from a slower introduction to programming

>>> CSC 108 will be offered in both Fall 2024 and Spring 2025



What is Computer Programming?

a.k.a., Software Development

Specifying step-by-step instructions for computing an answer, solving a problem or completing a task

Similar to a cooking recipe

- a sequence of activities to follow to cook something

- the sequence must be followed to obtain correct results (dish)

We save the instructions in a *program* so that we don't have to retype the commands every time

- Can run a program many times; each time same steps are taken



Often Encountered: Initial Difficulties in Learning to Program

Using a professional programming language (Java, C++, Basic) in an introductory programming course can be difficult, since two new skills need to be applied simultaneously

Imagine attempting to instruct a child who has never cooked before and who only speaks Portuguese to bake a cake (and you don't speak Portuguese!)

Two constraints must be faced at the same time:

- Must specify all actions exactly & explicitly, using simple, basic steps

- Must use a foreign language that you've never used before, spelling every command correctly and using the proper punctuation, etc.

This difficulty is faced by all CS departments in intro. courses



Our Approach to Introductory Programming at Mount Union

We have had success in our department with students of varying levels of prior programming experience in teaching the introductory programming course

Mount Union was a very early adopter of Java for our programming courses
first used here in 1996

We've changed our approach to the intro course many times, to make things better for students learning programming techniques

We've found that what works best for our students is an early and exclusive focus on **objects as everything**

Challenge: there aren't any textbooks available that use this exact approach

our solution? assigning appropriate readings from different texts at different times in the semester, from the University Library's O'REILLY LEARNING subscription

free full-text access for Mount Union students to over 50,000 titles!!



Objects as Everything? What are Objects?

Data in a computer program must be stored in the computer's memory

- keystrokes in a word processing document
- numbers entered into a spreadsheet
- images in a photo editor

Different programming languages have different ways of naming and accessing the data being processed by the program

The most widely used languages in professional software development today use **Object Oriented** data representations



Representing Data with Objects

Anything from the problem domain that is to be represented in a program is stored as an object

- the problem domain is the part of the outside world that is relevant to the program being written
- examples of object types -- TVSet, Automobile, Student

Objects have:

- a unique NAME
- one or more PROPERTIES (data values)
- one or more METHODS (capabilities)



Object Representation of a TVSet

There can be multiple TVSets in a house

- each one is given a unique name, for ease of identification
- “Hey, turn off that TV”
- “Which one do you mean? The Living Room TV or the Kitchen TV?”

Each object has a unique name, and a **class** that it comes from

- a class is the type of object we are talking about
- a general description of an object that could exist, describing the properties and methods of the object
- a class definition defines the structure of all objects that come from the same class
 - e.g., all students have the same properties and methods



Object Representation of a TVSet

A TVSet object has properties that represent values that describe the current state of operation of the TVSet

- can you think of some example properties that a TVSet might have?

A TVSet has methods that represent operations or things that the TVSet can do

- what kinds of things can you “ask” a TVSet to do with a remote control?



Object Representation of a TVSet

Some TVSet properties:

- channel
- volume
- brightness
- contrast
- on/off status

Some TVSet methods:

- channelUp
- volumeDown
- setChannel
- powerOn/Off



Objects as Everything

A pure Object Oriented programming language uses the Object concept to represent all data in a program

Java is an Object Oriented language that was developed from older, non-OO languages

- for ease of adoption by programmers who knew other languages, some non-OO features were built into Java

Most (all?) Java programming textbooks don't use a pure OO approach



Objects as Everything

But at Mount Union, we've found that using objects exclusively to represent everything in a program works best for teaching programming techniques

Everything in our Java programs is an object (including the program itself!)

Every program has properties (data) and methods (procedures and/or functions that can be called upon and carried out)



Objects as Everything Programming Approach

Our programs are collections of class definitions

We define the data objects that the program will contain

Then we ask the objects in our program to carry out operations in a specific sequence

an example: a program that displays several
Balloons on the screen...

CSC 120

Introduction to Programming with Java



Lecture 1

University of Mount Union
Math/Computer Science Department