

Programming 2A: Procedural Programming

Congratulations; you're speaking a different language! A programming language that is. But do you want to learn more? Discover the most popular programming languages and what they have to offer the software world. Explore data, algorithms, and objectives and how they are essential to language 'speak'. Learn the software development life cycle and how it can be implemented so you can create projects, such as a prototype for an app you'll code and a working to-do list website.

Companion courses listed at the bottom.

Unit 1: Hardware and Software Basics

Hardware and software form the perfect team; one without the other is useless. In this unit, we'll investigate how they work together. Tech projects usually list hardware and software requirements, so you will get to put on your project manager hat and identify those requirements! Then we will compare different types of software and when they should be used. We'll take a peek at the tiniest components of hardware and understand how they function. And you'll also learn tips for effective file structures and get some practice manipulating Python files. Get ready to zoom into the hardware and software world!

What will you learn in this unit?

1. Identify and describe the functionality of major software applications
2. Compare and contrast different operating systems
3. Create appropriate hardware and software requirements for a tech project
4. Write Python code that manipulates text files

Unit 2: Speaking the Same Language

How many languages do you speak? How many *programming* languages do you speak? There are a variety of different programming languages that fill different needs based on the type of software or website being developed. We are going to learn about the most popular programming languages and what they have to offer the software world. We will explore multitasking, exception handling, functions, and scope in Python. Get ready to learn another language!

What will you learn in this unit?

1. Compare and contrast different programming languages
2. Write Python programs that validate arguments and handle exceptions
3. Use global and local scope appropriately
4. Operate a debugging tool in Python

Unit 3: Working with Data

Data is all around us! Whether we realize it or not, every time we interact with a computer, we are generating lots of data. But data is not very useful until it is structured and processed. We are going to learn how data is represented by computers, how we can use data structures in Python, and how to validate data that is inputted by the user. We'll also learn about how data is processed, analyzed, and managed. Displaying data in a visual way can help users understand it better, so we'll examine some tips and tricks for data visualization. And we'll end with some practical Python data visualization.

What will you learn in this unit?

1. Identify and use a variety of data types and structures in Python
2. Write programs that input and validate user data
3. Compare techniques for data analysis and data processing
4. Describe how data is represented in binary

Unit 4: Introducing: Algorithms!

Algorithms are an important part of our daily lives, whether we realize it or not! We use algorithms whenever we follow a recipe or use step-by-step instructions. But algorithms can be very powerful and complex. We are going to examine several different classical algorithms that will help us search and sort data. We'll check out various data structures and learn some logic tips and tricks. We are even going to create a simulation of wasp breeding behavior—buzz!

What will you learn in this unit?

1. Describe and implement standard searching algorithms
2. Explain and implement standard sorting algorithms
3. Develop a Python program that implements an abstract data type
4. Understand and use Boolean algebra and De Morgan's Law

Unit 5: Algorithms in Action

Can you remember a day when you were feeling particularly efficient? Did you clean your room in record time? Did you complete a bunch of homework in an afternoon? Did you make dinner for the family without making the kitchen look like a disaster zone? We have all (hopefully!) experienced efficiency. We can describe algorithms in terms of efficiency as well. We'll learn a special type of notation that will help us evaluate an algorithm's time and space efficiency. We will also get plenty of practice writing and implementing a variety of algorithms along the way. Some might even help you with your math homework. Now that's efficient!

What will you learn in this unit?

1. Develop algorithms to solve common math problems
2. Identify and use recursion in programming solutions
3. Convert among binary, hexadecimal, and decimal systems
4. Understand and use Big-O notation

Unit 6: What's the Big Objective?

Objects are hugely important, both in real life and in the programming world. Your favorite objects in real life might be your phone, a pet, a drawing, a robot, or something else! An object in programming is modeled after real-life objects, and it helps us organize our code around the real world. The realm of object-oriented programming is vast and exciting, so buckle up for a wild ride! We're going to explore the different types of relationships between classes and objects. We'll also play with some speedy turtles that do whatever we tell them. Get your fingers warmed up because we are going to be typing a lot of code in this unit!

What will you learn in this unit?

1. Understand and implement composition, inheritance, encapsulation, and abstraction
2. Create a variety of classes and instantiate objects
3. Explain the differences between structural, object-oriented, and event-driven programming
4. Execute a variety of operations on objects including copying, comparing, and accessing/mutating

Unit 7: The Software Development Life Cycle

Writing code is just one small part of the process of developing software. To be successful in today's tech climate, projects need organization, management, and appropriate methodologies. We are going to explore the software development life cycle and the different ways that it can be implemented. We will design and code a small project that tracks inventory in a house. We'll end by examining a text adventure Python program that will give us more experience with large programs. Let's go!

What will you learn in this unit?

1. Compare and contrast design methodologies
2. Translate program design into appropriate data structures and code
3. Analyze, modify, and evaluate large programs
4. Discuss and apply various parts of the software development life cycle

Unit 8: Real-World Inspiration: Capstone Project

The real world is where the rubber meets the road, so to speak. It is where all the theory, concepts, and abstract ideas become practical. In this unit, we are going to create a prototype for an app that we will then code using the Flask framework. By the end of the unit, you will have a working to-do list website!

What will you learn in this unit?

1. Develop a multi-program application that uses real-world data
2. Use configuration management tools such as source control
3. Design a prototype using proper prototyping techniques
4. Create and test a capstone project using Flask in PythonAnywhere

Companion courses:

Principles of Information Technology

Introduction to Programming

© eDynamic Learning ULC | All Rights Reserved.