

Unit 4—Python 2...

HCCS STEM—Computer Science, IT, Engineering, Food/Nutrition, Business Studies

What do I need to be able to do?

I can **consolidate all Python concepts**, including sequences, selection, iteration, lists/tuples, and functions. This means I can bring together my understanding of how Python executes instructions sequentially, how to make decisions using if/elif/else, how to repeat actions with for and while loops, how to manage collections of data using lists and tuples, and how to define and use functions, both with and without parameters and return values.

I can **apply these concepts to build a small, functional program**. This involves designing and implementing a complete Python program by integrating fundamental programming constructs such as input/output, conditional logic, repetition, data structures, and modular functions to achieve a specific programmatic goal, such as a quiz game or a calculator.

I can **practice problem-solving and debugging**. This entails systematically identifying and resolving errors in my code, understanding the logical flow of a program, and using various techniques to test and verify code functionality and correctness.

Sequence refers to the execution of instructions one after another, in the precise order they are written, from top to bottom. This means each line of code is processed sequentially unless explicitly altered. For instance, a program that first prints a greeting and then asks for a user's name demonstrates this, as the greeting will always appear before the name prompt.

Selection is a programming construct that enables decision-making within the code, allowing a program to choose between different execution paths based on certain conditions. This is primarily achieved using if, elif (else if), and else statements. For example, a program can use selection to determine if a score is passing or failing, or if a word is "long" or "short" based on its length. It's vital for programs to respond differently based on user input or various conditions.

Your assessment is a 50 question examination paper. For this assessment, you must be prepared to answer questions on all the skills and code and give examples of each. The final few weeks require you to code independently.

Keywords

Sequence: Instructions are followed one after another, in the exact order they are written, from top to bottom.

Selection: A programming construct that enables decision-making in code using statements like if, elif, and else.

Iteration: The process of repeating actions, which can be either definite (using for loops where the number of repetitions is known) or indefinite (using while loops, continuing as long as a condition is true).

List: An ordered collection of items that is **mutable**, meaning its contents can be changed after it is created.

Tuple: An ordered collection of items that is **immutable**, meaning its contents cannot be changed once it has been created.

Function (Subprogram): A reusable block of code, defined using the def keyword, designed to perform a specific task or set of actions.

Parameter: Variables listed inside the brackets when you define a function, used to pass information into that function.

Return: A keyword used in a function to send a specific value back to the part of the code that called it, causing the function to immediately stop.

Procedures (also known as subprograms) and functions are reusable blocks of code designed to perform specific tasks, enabling problems to be broken down into smaller, more manageable units. Both are defined using the def keyword. The core distinction lies in their output: a **procedure (subprogram)** performs a specific task but **does not send a specific value back** to the part of the program that called it. An example of a procedure might simply print something to the screen. In contrast, a **function** performs a task and **calculates or retrieves a value, then sends it back** using the return keyword, which also causes the function to immediately stop. You need to be able to **define** and **call** these blocks of code. Functions can also be designed to receive **parameters**, which are variables listed when defining the function to pass information into it.

In Python, lists and tuples have a key difference in mutability. Lists are mutable, meaning their contents can be changed after creation, for data like a shopping list where items are frequently added, removed, or updated. A key operations for lists is accessing individual items using zero-based indexing, adding items with append() to the end or insert() at a specific position, removing items using remove() or pop(), determining their length with len(), and extracting sub-sections using slicing. In contrast, tuples are immutable and cannot be changed, which makes them suitable for data that will remain constant.

10 Test your understanding questions

Here are 10 multiple-choice questions summarising the whole unit:

1. What defines a "sequence" in Python programming?
A) Instructions that run randomly. B) Instructions that loop continuously. C) Instructions that are followed one after another, in the exact order they are written, from top to bottom. D) Instructions that depend on conditions.
2. Which Python keywords are primarily used for decision-making, allowing a program to choose different execution paths based on conditions?
A) for, while B) print(), input() C) if, elif, else D) append(), remove()
3. A for loop in Python is typically used for what type of iteration?
A) Indefinite iteration, where the number of runs is unknown. B) Definite iteration, where the exact number of times it will run is known before it starts. C) Conditional iteration, where it stops only if a specific condition is met. D) Random iteration, for unpredictable repetitions.
4. When is a while loop the most appropriate choice for iteration?
A) When you know precisely how many times the loop should execute. B) When you need to iterate through a fixed collection of items. C) When the loop needs to continue as long as a certain condition is true, and the number of repetitions is not necessarily known beforehand. D) When you want the loop to run only once.
5. What is the fundamental difference between a Python list and a Python tuple?
A) Lists are used for numbers, while tuples are used for text. B) Lists are unordered, while tuples are ordered. C) Lists are mutable (their contents can be changed), while tuples are immutable (their contents cannot be changed after creation). D) Lists use parentheses (), while tuples use square brackets [].
6. If you have a list `my_list = ["apple", "banana", "cherry"]`, how would you access the item "apple"?
A) `my_list(1)` B) `my_list` C) `my_list` D) `my_list("apple")`
7. To add an item to the end of a list or remove a specific value from a list, which methods would you use respectively?
A) `insert()` and `pop()` B) `add()` and `delete()` C) `append()` and `remove()` D) `extend()` and `slice()`
8. What is the primary difference between a procedure (subprogram) and a function in Python?
A) Procedures are defined with `def`, while functions are defined with `func`. B) Procedures always return a value, while functions never do. C) A procedure performs a task but does not send a specific value back, whereas a function performs a task and sends a value back using `return`. D) Procedures cannot take parameters, but functions can.
9. What is the primary difference between a procedure (subprogram) and a function in Python? A) Procedures are defined with `def`, while functions are defined with `func`. B) Procedures always return a value, while functions never do. C) A procedure performs a task but does not send a specific value back, whereas a function performs a task and sends a value back using `return`. D) Procedures cannot take parameters, but functions can.

10 Test your understanding answers

1. C, 2. C, 3. B, 4. C, 5. C, 6. C, 7. C, 8. C, 9. C