LO2: Understand how programming is implemented.

*2.1 Interpret variables within programming languages.*  
*2.2 Interpret common programming control structures that are used when developing code.*  
*2.3 Describe the use of common data structures.*  
*2.4 Describe how algorithms are used in programming.*  
*2.5 Describe how to test and debug programs.*

Don’t forget images!

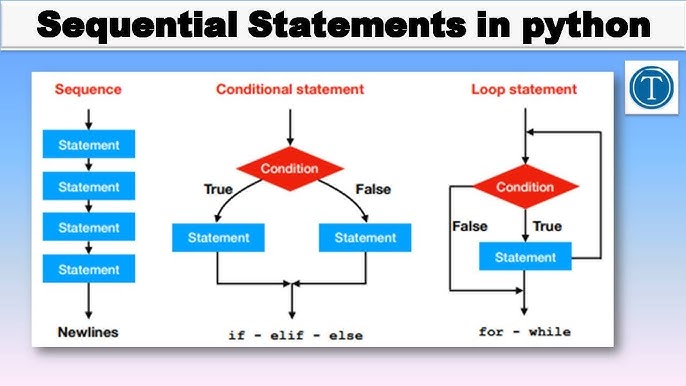
# Implementation

jfeiophvueopf

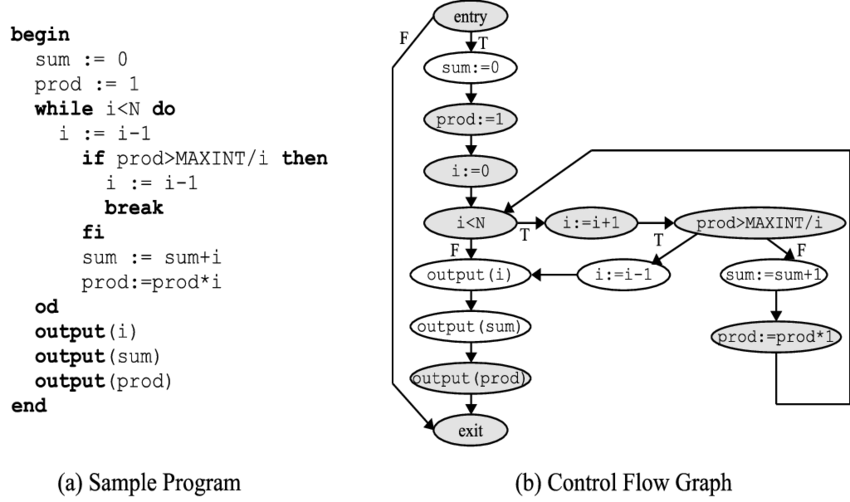
## Variables

fafasdsadad

## Flow Control



Dasdsadadas



### Sequencing

In programming, sequence is a basic algorithm (more on them later): A set of logical steps carried out in order. Computers need instructions in the form of an algorithm in order to complete a desired task, and this algorithm must have the correct order of steps, or sequence. Example: We can relate sequence to our everyday lives.

### Branches

Fsdfsfsdfsd

### Iteration

In your report, create a new section called ‘Implementation’.

In this section, explain with examples the following concepts:

* Variables: integer, float, Boolean, string, character.
* Simple programs that show sequencing
* Use of branches (if...then...else, elseif, case/switch
* Iteration (for next loops, while repeat etc)

# Data Structures

Add a new section called ‘data structures’ and then explain, with examples, the following:

* Arrays
* Linked lists
* Stacks
* Queues

## Arrays

Gsdgdsgsdffdsfdsff

hipHipArray = [“hip”,”hip”];

## Linked Lists

Fdsfdsfdsfdsfsd

https://www.linkedin.com/pulse/singly-linked-lists-understanding-real-life-examples-infant-regan-m7vnc/

https://www.geeksforgeeks.org/linked-list-implementation-in-c-sharp/

**Example of Use: Undo Feature in Apps**

Many applications, like text editors, use singly linked lists to implement the "Undo" feature. Each action is stored as a node in the list, and when you undo, the program navigates back through each node sequentially. This approach keeps memory usage efficient and ensures that actions are easily reversible.



## Stacks

fdsfdsfsdfs

## Queues

fdsfsdfsdf  
Algorithms

Add another new section called ‘Algorithms’.

Explain the concept of an algorithm and then give examples of typical algorithms including examples of:

* Sort algorithms (e.g. bubble, quick, insertion, merge)
* Search algorithms (linear, binary)

## Sort Algorithms

Write a simple one

Compare some common ones : complexity, effiency, speed

Code and illustrations

## Search Algorithms

Write a simple one

Compare some common ones : complexity, effiency, speed

Code and illustrations

# Debugging

Explain the concept of testing and, using an example program you have written, show how a software program can be tested in different ways such as:

* Use of debugging tools, break points, steps

Watches, conditionals, call stack

# Testing

Explain the concept of testing and, using an example program you have written, show how a software program can be tested in different ways such as:

* Use of debugging tools, break points, steps
* Black box and white box testing
* Explain the concepts of alpha and beta testing.

## Black Box and White Box Testing

## Alpha and Beta Testing

Alpha and beta testing share a common goal of identifying issues before product release. However, they differ in key ways. Alpha testing happens in-house, in a controlled environment, where developers focus on fixing critical bugs. Beta testing involves real users in real-world conditions.

alpha releases are initial versions of the software, used internally by developers to spot and correct bugs. Beta releases are then shared with a select audience outside the development team for further testing and feedback.

Alpha typically outside of the dev team