



Highsted Knowledge Organiser

High Performance Learning

Year 11: Computer Science: File Handling

What I need to know – Overview

- Understand what file handling is
- Identify different types of files
- Understand different file operations
- Write simple file handling Python programs
- Evaluate the use of file handling

Key Vocabulary (with definitions)

open ()	This function is used to open an existing or create a new file.
".txt" or ".csv"	This refers to the file name and location.
write ()	This function is used to write contents to the file.
read ()	This function is used to read contents from the file

Knowledge

File Handling Operations

File handling operations are needed so that a program can store and retrieve data permanently. Without file handling operations such as open, read, write, append, programs would not be able to save data. File handling is essential for applications like saving a game progress, storing user accounts and keeping logs.

Creating a file and Writing to the file - 1

newFile("names.txt"."w")	- Creates a file if it doesn't exist.
namesFile =	
open("names.txt")	- Open the file before writing to it.
namesFile.writeLine("Fred")	- Writing to file. Data is entered on a new line.
namesFile.writeLine("Laura")	
namesFile.close()	- Close the file after writing to it.

Creating a file and Writing to the file - 2

newFile("names.txt"."w")	- The data is stored to separate variable (name1, name2, name3)
namesFiles =	
open("names.txt")	
name1 = "Julie"	
name2 = "Mitch"	
name3 = "Aurora"	- This method allows the program to write each value to the file individually in the correct order
namesFile.writeLine(name1)	
namesFile.writeLine(name2)	
namesFile.writeLine(name3)	
namesFile.close()	

Reading data from the file

namesFile = open("names.txt"."r")	- The file is opened
name1 = namesFile.readLine()	- Read one line from file each time
name2 = namesFile.readLine()	
name3 = namesFile.readLine()	
print (name1)	- Store (print) in separate variables on different lines
print (name2)	
print (name3)	
namesFile.close()	

endOfFile

endOfFile() is used to determine the end of the file.

namesFile =	
open("names.txt"."r")	- Print out the contents using a while loop rather than reading from each line individually.
while not	
namesFiles.endOfFile()	- Loops are used as it allows the program to handle any number of data quicker and reduce errors
print(namesFiles.readLine())	
endwhile	
namesFile.close()	

Append data to a file

namesFile =	
open("names.txt", "a")	- Opens a file in append mode
namesFile.writeLine("Liam")	- Add new data to the end of the file
namesFile.close()	

File error handling

File not found	File doesn't exist
End of file error	Trying to read more data but reached end of file
Access/permission error	Program cannot read or write to file

Key Assessment Information

- There will be a written assessment on this topic at the end of term on how to create a file and access data from the file.