Hyperion Development

Task: Thinking Like a Programmer - Pseudo Code I

www.hyperiondev.com

Introduction

Welcome to the Pseudo Code Task!

This task will introduce the concept of pseudo code. Although pseudo code is not a formal programming language, it will ease your transition into the world of programming. Pseudo code makes creating programs easier, because as you may have guessed, programs can sometimes be complex and long, therefore, preparation is key. It is difficult to find errors without understanding the complete flow of a program. This is the reason why you will begin your venture into programming with pseudo code.

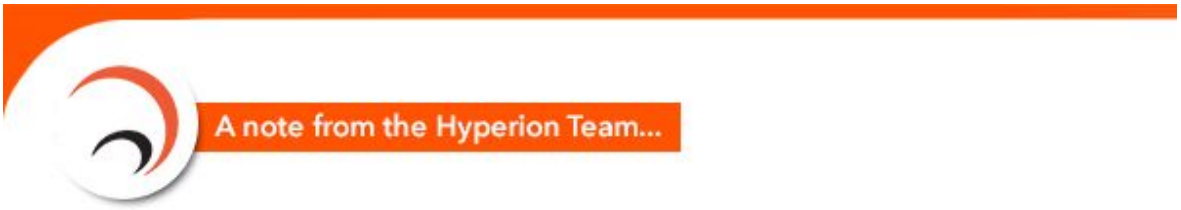
Connect with your mentor

CONNECT

Remember that with our courses, you're not alone! You can contact your mentor to get support on any aspect of your course.

The best way to seek help is to drop your mentor a line in the *comments.txt* file in your Dropbox folder. You can also log into www.hyperiondev.com/support to start a chat with your mentor, or schedule a 1:1 call with them.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or seek additional support!



Chances are, if you're taking this course, you're brand new to the Python programming language. You may even be new to programming altogether. Either way, you've come to the right place!

The key to becoming a competent programmer is utilising all available resources to their full potential. Occasionally, you may stumble upon a piece of code that you cannot wrap your head around or overcome. So, knowing which platforms to go to for help is important!

Hyperion knows what you need, and this is why we have a Slack group for students of our courses to assist each other. (And, of course, there are other avenues you can explore for help. One of the most frequently used by programmers around the world is Stack Overflow).

Also, our blog is a great place to find detailed articles and tutorials on concepts into which you may want to dig deeper. For instance, after being introduced to machine learning fundamentals here, you may want to read up further about machine learning on our blog, or learn how to deploy a machine learning model with Flask.

Our blog is updated frequently, so be sure to check back from time to time for new articles or subscribe to updates through our Facebook page.

Introduction to Pseudo code

What is pseudo code? And why do you need to know this? Well, being a programmer means that you will often have to visualise a problem and know how to implement the steps to solve that particular conundrum. This is where pseudo code comes in.

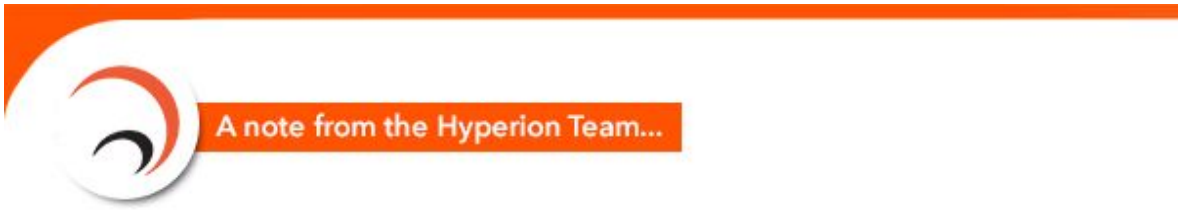
Pseudo code is a detailed yet informal description of what a computer program or algorithm must do. It is intended for human reading rather than machine reading and is therefore easier for people to understand than conventional programming language code. It also does not obey any specific syntax rules, unlike conventional programming languages, and therefore can be understood by any programmer, irrespective of programming language. As a programmer, pseudo code will help you better understand how to implement an algorithm, especially if it is unfamiliar to you. You can then translate your pseudo code into your chosen programming language.

Pseudo code is often used in computer science textbooks and scientific publications in the description of algorithms so that all programmers can understand them, irrespective of their knowledge of specific programming languages. It is also used by programmers in the planning of computer program development to initially sketch out their code before writing it in its actual language.

What is an algorithm?

Now, what exactly is an algorithm? Well, Newton's method is an example of an algorithm: it is a mechanical process of solving a category of problems. It is not easy to define an algorithm. It might help to start with something that is not an algorithm. When you learned to multiply single-digit numbers, you probably memorised the multiplication table. In effect, you memorised 100 specific solutions. That kind of knowledge is not algorithmic. But if you were "lazy," you probably cheated by learning a few tricks. For example, to find the product of n and 9 (when n is less than 10), you can write *n – 1* as the first digit and *10 – n* as the second digit. This trick is a general solution for multiplying any single-digit number by 9. That's an algorithm! Similarly, the techniques you learned for addition with carrying, subtraction with borrowing, and long division are all algorithms.

One of the characteristics of algorithms is that they do not require any intelligence to carry out. Algorithms are mechanical processes in which each step follows from the last according to a simple set of rules. In other words, an algorithm is like a recipe. It is a step by step method for solving a problem.



Did you know that Ada Lovelace (who was an English mathematician and writer born in 1815) is regarded as the first ever computer programmer? Her notes on an early mechanical, general-purpose computer (the Analytical Engine) is recognised as the first algorithm (pseudocode) to be carried out by a machine.

Ada Lovelace

Also, did you know the word 'algorithm' comes from the medieval Latin word 'algorism' and the Greek word 'arithmos' (ἀριθμός). The word 'algorism' (and therefore, the derived word 'algorithm') comes from Al-Khwārizmī (Persian: خوارزمی, c.780–850), a Persian mathematician, astronomer, geographer and scholar. The English language adopted the French term, but it wasn't until the late 19th century that "algorithm" took on the meaning that it has in modern English.

In English, it was first used about 1230 and then by Chaucer in 1391. Another early use of the word is from 1240, in a manual titled Carmen de Algorismo composed by Alexandre de Villedieu.

It begins thus:

> Haec algorismus ars praesens dicitur, in qua / Talibus Indorum fruimur bis quinque figuris.

which translates as:

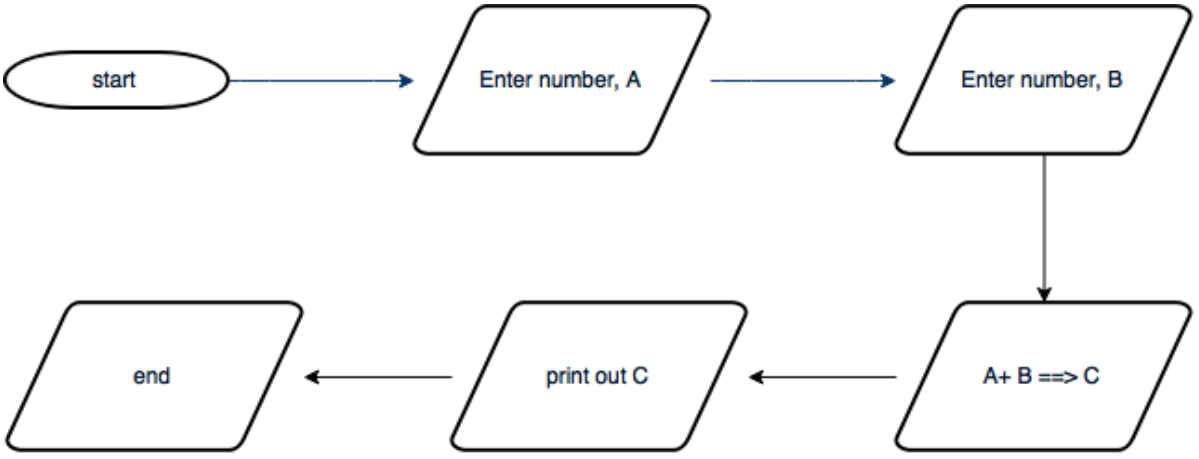
> Algorism is the art by which at present we use those Indian figures, which number two times five.

The poem summarises the art of calculating with the new style of Indian dice, or Talibus Indorum, or Hindu numerals or al-adad al-Hindi.

Pseudo code Syntax

There is no specific convention for writing pseudo code, as a program in pseudo code cannot be executed. In other words, pseudo code itself is not a programming language, rather is a model of all programming languages which one uses to make eventual coding a little simpler. Pseudo code resembles programming syntax, but pseudo code cannot be compiled without errors.

Flowcharts can be thought of as a graphical implementation of pseudo code. This is because a flowchart is more visually appealing and probably more readable than a "text-based" version of pseudo code. For example, for an algorithm that prompts a user to enter two integers consecutively, and then proceeds to sum up the two inputs and print out the result, would look something like this:



Instructions

First read **example.py** in this task folder. Open it using Notepad (Right-click on the file and select 'Edit' or select 'Open with' and then 'Notepad').

- example.py should help you understand some simple pseudo code. Every task will have example code to help you get started. Make sure you read all of the example file(s) and try your best to understand.

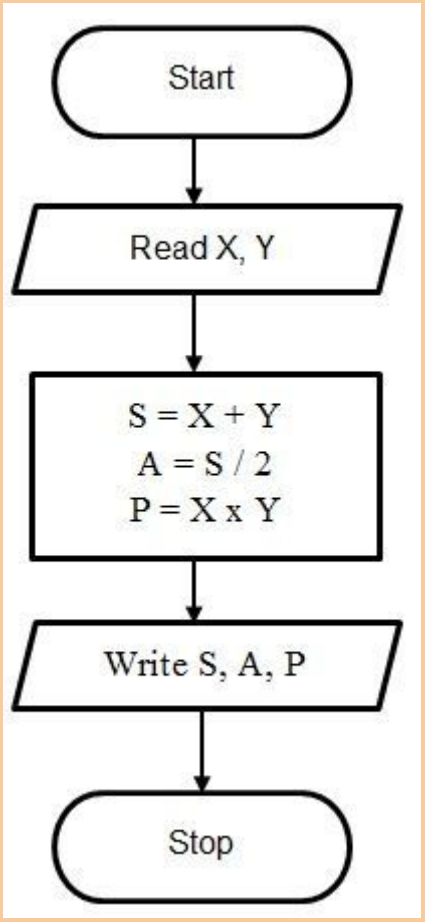
- This may take some time at first as it is slightly different from other languages, but persevere! Your mentor is here to assist you along the way.

- Feel free to write your own example code before doing this task to become more comfortable with some of the basic components.

Compulsory Task

Follow these steps:

- Create a new text file called **pseudo.txt** inside this folder.
- Inside **pseudo.txt**, write pseudo code for the following scenarios:
    - An algorithm that asks a user to infinitely enter a positive number until the user enters a zero value, and then determines and outputs the largest of the inputted numbers.
    - An algorithm that reads an arbitrary number of integers and then returns their arithmetic average.
    - An algorithm that reads a grocery list and prints out the products (in alphabetical order) that are still left to buy.
    - An algorithm for the flowchart below:



Things to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python or Notepad++** may not be installed correctly.
2. If you are not using Windows, please ask your mentor for alternative instructions.

Share your thoughts

RATE

Hyperion strives to provide high quality, relevant course content that helps you achieve your learning outcomes. Think the content of this task (or this course as a whole) can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.