

A-Level Computer Science P2 Notes (Python)

First Edition

Navid Saqib

(0333-4259883)

Visiting Teacher

Lahore Grammar School

Beacon House School

SISA

KIMS

ROOTS

All rights reserved. No part of this publication may be reproduced, Stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher.

| | |
|-----------------------------|---|
| Title | A-Level Computer Science P2 Notes (Python) |
| Author | Navid Saqib. (0333-4259883) |
| Published by | MS Books (042-35774780) |
| Edition | First Edition |
| Legal Advisor | Ashir Najeel Khan (Advocate High Court) AKBAR LAW CHAMBERS 39-40, 1 st Floor, Sadiq Plaza, The Mall, Lahore 042-36314839, 0307-4299886 |
| For Complaints/Order | MS Books 83-B Ghalib Market, Gulberg III Lahore (042-35774780),(03334504507),(03334548651) |

CONTENT TABLE

| Ch # | Topics | Pg # |
|-------------------------|--|------|
| 1. | Knowing Algorithm and Pseudocode | 5 |
| 2. | Identifier Table | 6 |
| 3. | Programming stage 2 Input | 9 |
| 4. | programming stage 4 output | 10 |
| 5. | Example of Totaling | 14 |
| 6. | Selection construct | 20 |
| 7. | Nested if statements | 33 |
| 8. | Iteration | 34 |
| 9. | 1D arrays | 47 |
| 10. | Finding maximum and minimum value of array | 52 |
| 11. | 2D array | 55 |
| 12. | Built in functions | 58 |
| 13. | Procedures | 67 |
| 14. | Functions | 75 |
| 15. | Passing Array to functions | 83 |
| 16. | Structured chart | 85 |
| 17. | Bubble sort | 89 |
| 18. | File Handling | 91 |
| 19. | Dry running and algorithm | 95 |
| PRACTICE BOOKLET | | |
| 20. | if/then/else/selectcase practice | 101 |
| 21. | Iteration loop practice 1 | 129 |
| 22. | Iteration loop practice 2 | 138 |
| 23. | 1D array practice | 145 |
| 24. | 2D array Practice | 155 |
| 25. | Random Number Generation practice | 164 |
| 26. | Procedure and Function Practice | 171 |
| 27. | File reading and writing Practice | 187 |

Knowing Algorithms and Pseudo codes

Algorithms:

An algorithm (pronounced AL-go-rith-um) is a procedure or formula for solving a problem.
Mostly written as English Statements

Or

An algorithm is a set of instructions designed to perform a specific task. This can be a simple process, such as multiplying two numbers, or a complex operation

Or

A set of instructions independent of any programming language that calculates a function or solves a problem

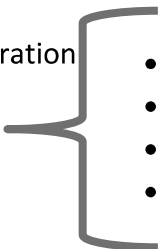
Algorithm is a sequence of steps involved to solve a problem. These steps are:

Pseudo code:

Informal high-level description of a computer program or other algorithm, intended for human reading rather than machine reading.

Pseudo code is not a programming language, but simply an informal way of describing a program. It does not require strict syntax, but instead serves as a general representation of a program's functions.

Main Steps of solving a problem/programming

- 
- Setting and declaration
 - Input
 - Process
 - Output
- Setting and declaration
 - Input
 - Calculations/constructs
 - Output

IDENTIFIER TABLE

- Deciding about ingredients Using IDENTIFIER(variable) TABLE

| Ingredient/Equipment Name | Quantity Needed | Description |
|---------------------------|-----------------|-----------------------------------|
| Tea | 1 Tea spoon | Required for the Flavor and color |
| Sugar | 2 Tea spoon | For sweetener |
| Water | 1 cup | For Diluting the substance |
| Milk | ½ cup | For enriches the taste |
| Kettle | 1 unit | For processing |
| Stove | 1 | For processing |
| Cup | 1 unit | For pouring tea |
| Saucer | 1unit | For cup |
| Spoon | 1 | For presentation and mixing sugar |
| Teapot | 1 | For making tea in it |

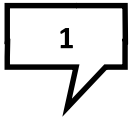
Solution to the problem step by step:

| | |
|--|---|
| Put kettle on stove Put Water in Kettle If it is an electric kettle then Switch the kettle on Else Put it on lit gas/turn stove on End if Setup the cup and saucer Put a spoon on the saucer Get the Teapot Put the tea in teapot If you take the milk then Add the milk to the cup Endif | If you take sugar then Add the sugar to the cup End if Check while water has not boiled Wait Endwhile Add water to teapot Wait for 3 minutes to let it BRU Pour tea into the cup If there is a sugar in the cup Stir the tea End if Repeat Wait Until tea is cool enough to drink Enjoy Tea! |
|--|---|

The solution given is a conceptual one not a programming code

Difference between

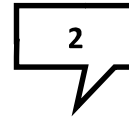
1. Algorithm
2. Pseudo code
3. Programming statements



Set total equal to zero



total = 0



total = int(0)

Programming Stage 1: **Setting and declaration**
Here is a table showing all the parameters

| Type | Description | Memory Space | Example |
|-----------|---|--|------------------------|
| Integer | a whole number from -2,147,483,648 through 2,147,483,647 | 4 bytes | 37,453 |
| Byte | a whole positive number from 0 to 255 | 1 byte | 12 |
| Real | Programming Languages does not use Real Numbers, instead it uses {Single} and {Double}, which both allow for decimal places | - | - |
| {Single} | 1.5×10^{-45} to 3.4×10^{38} | 4 bytes | 1002.375 |
| {Double} | 5.0×10^{-324} to 1.7×10^{308} | 8 bytes | 9997.775 |
| Decimal | 7.9228×10^{-28} to 7.9228×10^{28} | 16 bytes | 3.8 |
| Boolean | either TRUE or FALSE Alternatively 1 or 0 Alternatively Yes or No | 4 bytes (!) | TRUE |
| Character | A single character | 2 bytes | J |
| String | A collection of characters | A unicode string with a maximum length of 2,147,483,647 characters | Cabbage |
| Date/Time | There are several different types of date format that you can apply. 01/01/0001 to 12/31/9999 and times from 12:00:00 AM (midnight) through 11:59:59.9999999 PM | 8 bytes | 08/17/1924 14:34:23 |

Contain two element s

Variables – (changeable values) which can change during the execution of the program
e.g

Price, quantity, total, speed, no of students

Constant – (static Values) which cannot change their values during the program
e.g

Value of π which is 3.142 or 22/7

*In a program you can also create your own constant by declaring its value in the start of the program

How to declare a variable and a constant in programming language?

<Variable> = <data type><Value>

Num1 = int(0)

Here the variable of num1 has been declared this contains no initial value but it has to be input by the user

Pseudo code:

DECLARE num1 : INTEGER

DECLARE num2 : INTEGER

```
num1 = int(0)
num2 = int(0)|
```

NAVID SAQIB

Programming stage 2: INPUT

There are two components of input

- **User message**
- **Assigning value to variable/ obtaining value from user**

After declaring the variable now its turn to assign its value so that the variable can be used for further processing .To assign a value to variable, you must output a valid user message to the user so he/she can understand what type of input is required.

Algorithm

Num1 = INPUT "Input num 1"

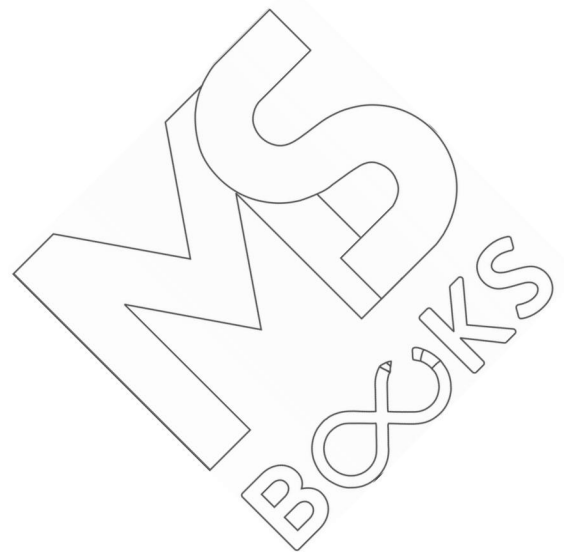
Num2 = INPUT "Input num 2"

Python Code

```
Q1.py - C:/Users/Navid Saqib/Desktop/Paper 2
File Edit Format Run Options Window |
num1=int(0)
num2=int(0)
num1=input("Enter Number 1 : ")
num2=input("Enter Number 2 : ")
|
```

This message will appear on console screen like this:

```
Python 3.6.6 Shell
File Edit Shell Debug Options
Python 3.6.6 (v3.6.6:4cflf.
1)] on win32
Type "copyright", "credits'
>>>
  RESTART: C:/Users/Navid S.
Enter Number 1 : 23
Enter Number 2 : 45
>>> |
```



Programming stage 4: OUTPUT

Displaying output is the one of the most important parts of a program. A program will be useless until and unless output is not properly displayed.

As you have read in INPUT stage output is displayed on console screen through command: **Print()**

- To output the value of a variable “,” is used right after the end of user message
- This “,” is known as Concatenation which joins the User message with value of variable.
- After “,” the variable in which value is stored is to be written.

The code is written like this:

```
print ("number 1 is :", num1)
```

This would output this on the console screen:

```
Enter num1 12
Enter num2 32
number 1 is : 12
>>> |
```

Algorithm :

OUTPUT “Product value is :”, num1

NAVID SAQIB