



PALAWAN STATE UNIVERSITY
College of Sciences



CC2/L

Computer Programming I

Module 1
General
Introduction





General Introduction

1.1 Introduction: The Way of the Program

The goal of this course is to teach you to understand and create programs in Python. We will outline the learning objectives that should be accomplished once you have gone through the chapter.

Computers do what you tell them to do, not what you mean for them to do. Thus understanding computer code involves a lot of mental simulation of what will actually happen, not what you wish would happen.

Writing computer programs requires not only mechanistic thinking but creative problem-solving. Problem-solving means the ability to formulate problems, think creatively about solutions, and express a solution clearly and accurately. As it turns out, the process of learning to program is an excellent opportunity to practice problem-solving skills. - Coursera



Learning Objectives

After going through in this module, you should be able to:

- ✓ Describe the purpose of programming languages.
- ✓ Apply the algorithm when solving problems.
- ✓ Execute a simple program in any python Integrated Development Environment (IDE).
- ✓ Identify basic program structures in Python.



Overview

1.2 Programming Language

A programming language is a formal language that is both

- universal (any computable function can be defined)
- Implementable (on existing hardware platforms)

It is a notation for writing programs, which are specifications of a computation or algorithm. Some authors restrict the term "programming language" to those languages that can express all possible algorithms. A programming language is a computer language programmers use to develop software programs, scripts, or other sets of instructions for computers to execute.

Although many languages share similarities, each has its own syntax. Once a programmer learns the languages rules, syntax, and structure, they write the source code in a text editor or IDE. Then, the programmer often compiles the code into machine language that can be understood by the computer. Scripting languages, which do not require a compiler, use an interpreter to execute the script.



Discussion

1.3 Algorithm

The whole world is digitalized today. There is a sense of intelligence; there is a sense of communication in every traditional device that makes our lives so easy, so fast. All these technological advancements are taken forward by software which is a bunch of programs that are meant to solve a problem. And every program is built upon a logic/solution, which is called as an Algorithm. The naming **algorithm** is named after the clever man from Baghdad, Al Khwarizmi. He was the first person to introduce algorithms to the world which were mechanical, precise and unequivocal.

An algorithm is a well-defined step-by-step solution or a series of instructions to solve a problem. It can be the method to find the least common multiple of two numbers or the recipe to cook buttered vegetables (EDUCBA, 2021).



Discussion

Algorithm

What is an algorithm from a programming perspective?

The computer basically does a lot of math which means it has a lot of problems to solve. That's exactly why algorithms form the heart of *computer science*. A computer algorithm is a computational procedure that takes in a set of finite input and transforms it into output by applying some math & logic. An algorithm in programming will have several steps as follows –

1. Problem definition – What is to be done?
2. Data collection – What do we have to solve the problem? Or inputs.
3. Data processing – Understanding what we have or transforming them into a usable form.
4. A logical approach – Employing the collected & created data against logic to solve.
5. Solution – Present the solution in the way you want in a GUI or a terminal or a diagram or a chart.



Discussion

Algorithm

Kinds of Algorithm

Kinds	Purpose
Brute force algorithms	Which are straightforward trial and error approaches to solving problems.
Divide and Conquer algorithms	This breaks the problem into small subproblems and then combines each subproblem result to get the final result.
Greedy Algorithm	This follows a problem-solving in experimental way to reach the next best solution to find the final best solution.
Dynamic Programming	An approach that is the same as divide and conquers but divides the problem into subproblems such that their results are reusable for other subproblems.



Discussion

Algorithm

Advantages of the algorithm, and why should we use the algorithm in programming?

Talking about why we should use algorithms in programming, we must understand that computer programs adopt different algorithms that run on computer hardware that has a processor & memory, and these components have limitations. A processor is not infinitely fast, and the memory we have is not free. They are bounded resources. They must be used wisely, and a good algorithm that is efficient in terms of time complexities and space complexities will help you do so.



Discussion

Algorithm

Characteristics of an good Algorithm

Not all procedures can be called an algorithm. An algorithm should have the following characteristics –

1. Unambiguous – Algorithm should be clear and unambiguous. Each of its steps (or phases), and their inputs/outputs should be clear and must lead to only one meaning.
1. Input – An algorithm should have 0 or more well-defined inputs.
1. Output – An algorithm should have 1 or more well-defined outputs, and should match the desired output.
1. Finiteness – Algorithms must terminate after a finite number of steps.
1. Feasibility – Should be feasible with the available resources.
1. Independent – An algorithm should have step-by-step directions, which should be independent of any programming code.



Discussion

Algorithm

How to Write an Algorithm?

There are no well-defined standards for writing algorithms. Rather, it is problem and resource dependent. Algorithms are never written to support a particular programming code.

We write algorithms in a step-by-step manner, but it is not always the case. Algorithm writing is a process and executed after the problem domain is well-defined. That is, we as programmer should know the problem first, for which the design of a solution should be aligned to solve a specific problem.



Initial Activity

Algorithm

Let's try to learn algorithm-writing by using an example.

Example 1:

Problem – Design an algorithm to add two numbers and display the result.

step 1 – START

step 2 – declare three integers a, b & c

step 3 – define values of a & b

step 4 – add values of a & b

step 5 – store output of step 4 to c

step 6 – print c

step 7 – STOP



Initial Activity

Algorithm

Algorithms tell the programmers how to code the program. Alternatively, the algorithm can be written as –

step 1 – START ADD

step 2 – get values of a & b

step 3 – $c \leftarrow a + b$

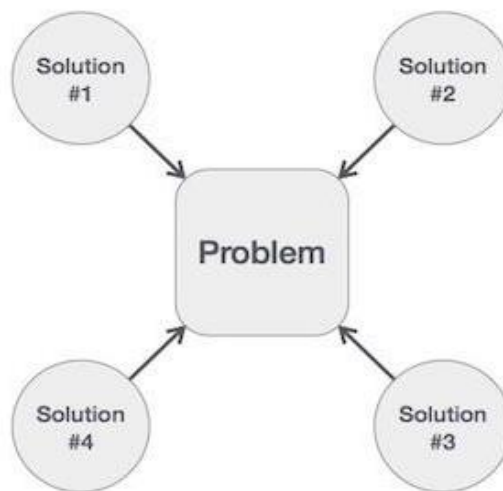
step 4 – display c

step 5 – STOP

Tips



In design and analysis of algorithms, usually the second method is used to describe an algorithm. It makes it easy for the analyst to analyze the algorithm ignoring all unwanted definitions. He can observe what operations are being used and how the process is flowing. Writing step numbers, is optional. We design an algorithm to get a solution of a given problem. A problem can be solved in more than one ways.



Hence, many solution algorithms can be derived for a given problem. The next step is to analyze those proposed solution algorithms and implement the best suitable solution.



Learning Check

Create an algorithm that will compute the:

- Area of a circle using the formula $A = \pi r^2$.
- Average of three numbers using the formula $\text{Average} = (n_1 + n_2 + n_3)/3$



Discussion

Python as Programming language is a high-level language, together with C++, PHP, and Java.

Why Python?

- take less time to write programs
- portable, they can run on different kinds of computer
- works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- has a simple syntax similar to the English language.
- has syntax that allows developers to write programs with fewer lines than some other programming languages.
- runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- can be treated in a procedural way, an object-oriented way or a functional way.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.



Discussion

How to get started with the Python Programming Language

Python Install

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on Linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: Download [PyCharm: Python IDE for Professional Developers by JetBrains](#)

Python QuickStart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.



Discussion

Integrated Development Environment: PyCharm

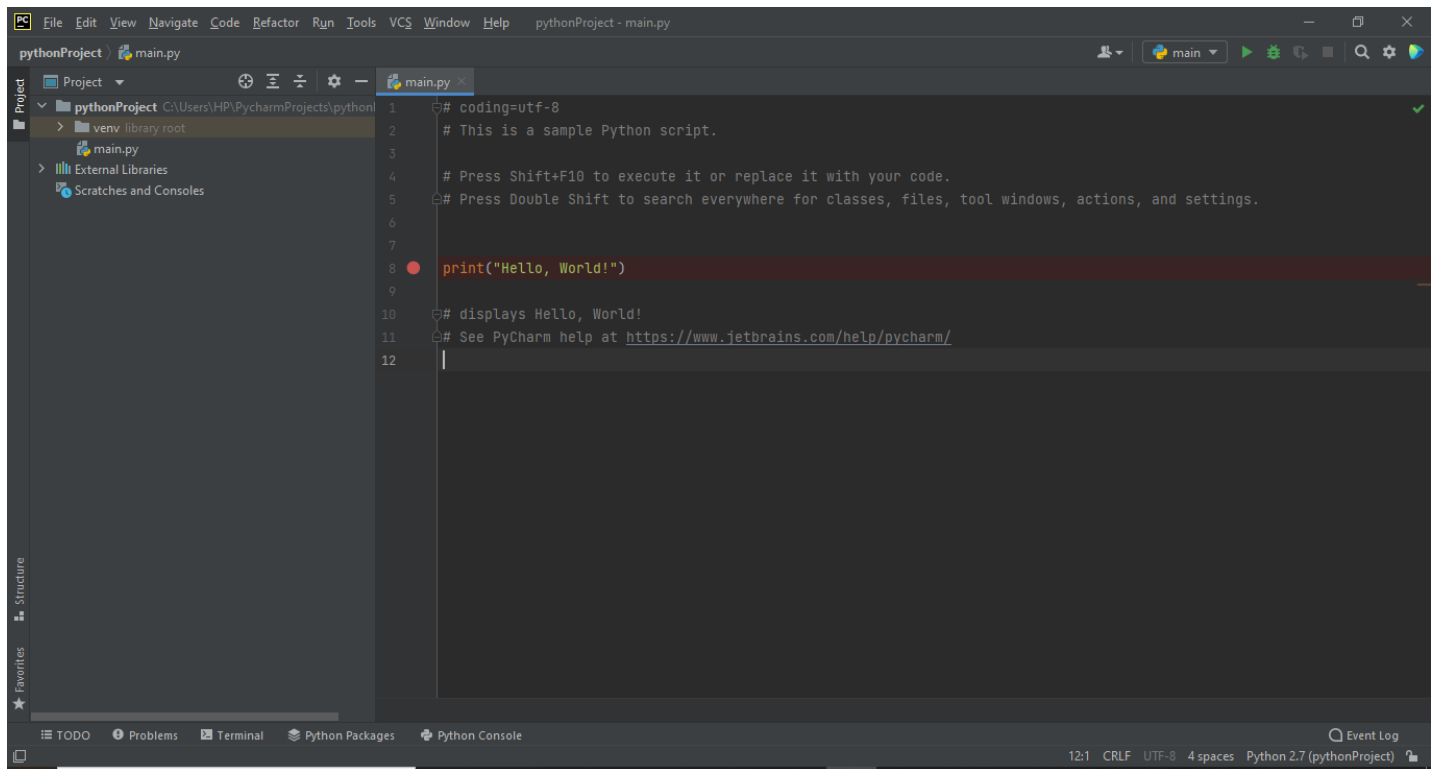
Integrated Development Environment (IDE) is a software application that provides comprehensive services to computer programmers for software development. An IDE normally consists of at least a source code editor, builds automation tools and a debugger.

PyCharm is a cross-platform IDE that provides consistent experience on the Windows, macOS, and Linux operating systems. It is available in three editions: Professional, Community, and Edu. The Community and Edu editions are open-source projects and they are free, but they have fewer features. PyCharm Edu provides courses and helps you learn programming with Python. The Professional edition is commercial, and provides an outstanding set of tools and features.



Discussion

IDE: PyCharm Environment





Initial Activity

Let's Try

Write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py  
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read: Hello, World!

Congratulations, you have written and executed your first Python program.



Initial Activity

Problem:

Create a python program that displays an output same as shown in the window below.

Note: Pycharm is being used as an editor in this sample.

Answer:

```
print ("Hello, World!")
```

```
print("Hello, World!")
```



Tips



TIPS

- When saving your program, use the file extension “.py”.
- # is used for writing comments to inform the programmer what the code’s purpose is.
- Python programming language is case and space sensitive
- PyCharm is one of the useful interpreters/editors for python programming language



Learning Check

Create a program that will display your personal details as a student of Palawan State University. Save the program as “Student_Infor.py”.

Sample Output:

Name: Your name

Birthdate: May 7, 2002

Address: Sta. Monica, Puerto Princesa City

Program: Bachelor of Science in Computer Science

Year/Block: First Year - Block 1

Department: Computer Studies

College: Sciences



Evaluation

EVALUATION

I. Multiple Choice: Choose the letter of the correct answer.

- A. The first sentence is correct. The second statement is incorrect.
- B. The first sentence is incorrect. The second statement is correct
- C. Both statements are correct.
- D. Both statements are incorrect.

- 1. Programming Language is a computer language programmers use to develop software programs, scripts, or other sets of instructions for computers to execute. It is a notation for writing programs, which are specifications of a computation or development.
- 2. An algorithm is a well-defined step-by-step solution or a series of instructions to solve a problem. It is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.

II. Application: Write the correct algorithm to do the tasks below.

- 1. Computes a volume of a sphere.
- 2. Computes an average of three quizzes of a student.



Rubrics

Rubrics in checking an algorithm

- 5 – Excellent
- 4 – Outstanding
- 3 – Satisfactorily
- 2 – Fair
- 1 – Poor

Criteria	5	4	3	2	1
1. Unambiguous - inputs/outputs which should be clear and must lead to only one meaning.					
2. Input – an algorithm should have 0 or more well-defined inputs.					
3. Output – An algorithm should have 1 or more well-defined outputs, and should match the desired output.					
4. Finiteness – Algorithms must terminate after a finite number of steps.					
5. Feasibility – Should be feasible with the available resources.					
6. Independent – An algorithm should have step-by-step directions, which should be independent of any programming code.					

Summary

Discuss each Acquired Knowledge in Learning Outcomes

- ✓ Programming Language is a special language or a set of instructions to communicate with computers.
- ✓ Programming language has a set of rules (like English has grammar) to follow and it is used to implement the algorithm to produce the desired output.
- ✓ An algorithm can be defined as a step-by-step procedure for accomplishing a task.
- ✓ An algorithm can be used to improve the speed at which a program executes a problem
- ✓ Python is a high-level programming language, meaning that it has a clear syntax that reads a lot like English
- ✓ Python can run on any machine, whether it's Mac, Windows, or Linux which means that it can be written and used across different systems without making any changes to the code.



References

EPN Enterprise Podcast Network. (2020, January 27). The Importance of Programming Languages. Retrieved from: <https://epodcastnetwork.com/the-importance-of-programming-languages/>

Goes, Mikke. (2020, October 9). 12 Essential Advantages of Python. Retrieved from: <https://mikkegoes.com/advantages-of-python/>

Mulongo, Cleophas. (2021, August 5). The Importance of Algorithm in Computer Programming. Retrieved from: <https://www.technotification.com/2019/02/importance-of-algorithms-programming.html>

Pedamkar, Priya. (2020) Algorithm in Programming. Retrieved from: <https://www.educba.com/algorithm-in-programming/>

Software Testing Help. (2021, August 27). Basics Of Computer Programming For Beginners | Coding Tutorial. Retrieved from: <https://www.softwaretestinghelp.com/basics-of-computer-programming/>

W3schools. (2021). Python Tutorial. Retrieved from: <https://www.w3schools.com/python/default.asp>