

Project Outline

Lesson 1: Introduction to Project

- Introduce Python programming language and its applications
- Discuss the importance of variables, input, and validation in programming
- Provide an overview of the simple program
- Discuss the Problem Solving process.

Lesson 2: Translators

- Compare how humans and computers interpret instructions
- Explain the differences between high- and low-level programming languages
- Describe why translators are necessary
- List the differences, benefits, and drawbacks of using a compiler, or an interpreter

Lesson 3: Sequence

- Describe the tools an IDE provides (editors, error diagnostics, runtime environment, translators)
- Use subroutines in programs
- Define a sequence as instructions performed in order, with each executed in turn
- Predict the outcome of a sequence and modify it
- Interpret error messages; define error types and identify them in programs (logic, syntax)

Lesson 4: Variables

- Use meaningful identifiers
- Determine the need for variables
- Distinguish between declaration, initialisation, and assignment of variables
- Demonstrate appropriate use of naming conventions
- Output data (e.g. `print (my_var)`)

Lesson 5: Input

- Obtain input from the keyboard in a program
- Differentiate between the data types: integer, real, Boolean, character, string
- Cast variables by calling a function that will return a new value of the desired data type
- Define runtime errors in programs
- Define validation checks

Lesson 6: Flowcharts

- Identify flowchart symbols and describe how to use them (start, end, input, output, subroutine)
- Translate a flowchart into a program sequence
- Design a flowchart for a program

Lesson 7: Planning and Designing the Simple Program

- Have students create a flowchart for the simple program
- Discuss the variables needed for the program and how to validate the user's input

- Have students identify the variables needed for the program and how to validate the user's input
- Have students document their flowchart and variable identification process

Lesson 8: Subroutines

- Describe a subroutine
- Describe the purpose of parameters in subroutines
- Use procedures that accept arguments through parameters
- Describe how subroutines are used for decomposition
- List the advantages of subroutines

Lesson 9: Implementing the Simple Program

- Introduce the code for the simple program
- Have students implement the code for the simple program
- Discuss how to use comments and documentation to make the code more readable and understandable
- Have students comment and document their code

Lesson 10: Data validation

- Determine the need for validation checks
- Use iteration to perform validation checks

Lesson 11: Testing and Debugging the Simple Program

- Introduce the concept of testing and debugging in programming
- Have students test their simple program and identify any errors or bugs
- Discuss how to handle errors and exceptions in the program
- Have students revise their code based on the testing and debugging process

Lesson 12: Improving the Simple Program

- Introduce the concept of improving a program to create a more robust program
- Have students identify areas for improvement in their simple program
- Discuss how to apply the iterative steps of Define, Prepare, Prototype, and Test when improving the program
- Have students revise their code to create a more robust program

Lesson 13: Reflection and Assessment

- Have students reflect on their learning and growth throughout the unit
- Assess students' understanding of Python programming, variables, input, and validation through a quiz or project
- Provide feedback to students on their work and identify areas for improvement