CCI Learning®
IGNITING POSSIBILITIES

Information Technology Series

# Python

## Level 1

## Instructor Guide



INFORMATION TECHNOLOGY SPECIALIST

# Python – Level 1
# Instructor Guide

# Overview

## Course Structure

The course is structured to learn by doing, practice the learned skill, and then apply the skills.

- Unit
  - Lesson
    - Learn Tasks
    - Practice Exercises
    - Practice Questions
  - Objective Assessment
  - Create Project

Each unit contains lessons. The lessons are introduced by lesson topics where learners can understand through doing or learning through study material. Each lesson concludes with a Practice Exercise that incorporates the tasks they learned throughout the lesson. Once they have completed the lessons in the unit, learners are assessed through a question-based Objective Assessment and a Create Project.

## Delivery

The course is created so it can be customized to meet the needs of the instructor and the learner.

- **Direct Instruction**: Utilize the PowerPoint Presentations to introduce each lesson topic, then have the learners review the study materials and complete the task.
- **Flipped Classroom**: Learners complete reading and exercises outside of class time. Learners utilize the class time to discuss learned tasks, allow learners to teach concepts, expand concepts through learning stations and work on unit extension or unplugged activities.
- **Learner-Centered Approach:** Learners can work at their own pace on their own schedule to complete the course. Instructors support learners by utilizing the answer keys to identify struggles and guide learners through the solutions.

## Prepare for Delivery

- Begin with the unit overview to understand the structure and flow of the unit, the topics covered, the approximate time to complete and the exam objectives reviewed.
- Review the lesson PowerPoint Presentation to give you an in-depth look at each lesson topic and the comprehensive topic notes included.
- Review the answer keys to familiarize yourself with the tasks learners will complete throughout the lesson.
- Complete the lesson.

# Instructor Resources Overview

| | |
|---|---|
| **Instructor Resources**<br><br>**File Structure** | Instructor Resources<br>Course Syllabus<br>Course Overview<br>Course Key Terms<br>Unit<br>Unit Assessment Answer Keys<br>Create Project<br>Objective Assessment<br>Lesson<br>Answer Keys<br>Lesson Practice Exercises<br>Learn Tasks<br>Lesson Practice Questions<br>Study Guides<br>Study Guide Complete<br>Study Guide Fill-In Explanation<br>Study Guide Fill-In Topic<br>Lesson PowerPoint Presentations<br>Unplugged Activities<br>Unit Overview<br>Unit Learning Plan<br>Unit Key Terms |
| **Unit Assessment Answer Keys** | Each unit includes two types of assessments for learners to apply their knowledge.<br><br>• **Create Project** – These are project prompts and sample solution files. Create projects also include "show me" videos in XperienceED for learner reference. You have the option to enable/disable this feature.<br>• **Objective Assessment** – A comprehensive question and answer-based assessment for the unit. Objective Assessments include "show solution" in XperienceED for learner reference. You have the option to enable/disable this feature. |
| **Answer Keys** | Documents containing answers, step-by-step instructions, and correct answers for Instructor reference or to offer additional support material for learners.<br><br>• **Learn Tasks** - Each lesson topic includes an opportunity to apply what they have just learned in-app or by answering questions. Learn Tasks also include "show me" videos and "show solution" in XperienceED for learner reference.<br>• **Lesson Practice Exercises** – End of lesson in-app or scenario-based assessment. Lesson Practice Exercises also include "show me" videos in XperienceED for learner reference. You have the option to enable/disable this feature. |

| | |
|---|---|
| | • **Lesson Practice Questions** – End of lesson question-based assessment. Lesson Practice Questions also include "show solution" in XperienceED for learner reference. You have the option to enable/disable this feature. |
| **Study Guides** | Printable and customizable study guides mapped to lesson topics and exam objectives are provided in three formats.<br>• **Complete**- This version includes the topic and the explanation.<br>• **Fill-In Topic**-Learners can fill in the topics as they learn or as a review.<br>• **Fill-In Explanation** – This allows learners to complete the explanation of each lesson topic in their own words and images. |
| **Learning Plan** | Customizable unit learning plan outlining the objectives and topics covered, essential questions, learning targets, methods and materials, extension activities, formative and summative assessments, mapping to STEAM, Work Readiness, 5 C's, and Bloom's Taxonomy Levels. |
| **Lesson PowerPoint** | A PowerPoint Presentation that complements the XperienceED lesson. Each lesson topic is included in the presentation as well as comprehensive speaker notes. |
| **Unplugged Activities** | A variety of activities and necessary resources to get learners off the computers while still reinforcing unit learning objectives. |
| **Unit Key Terms** | A comprehensive list of key terms throughout the unit. |
| **Unit Overview** | A spreadsheet containing the overview of the flow of the unit that includes lesson topics, certification objectives mapping, and approximate timings for self-paced and instructor-led scenarios. |

# Python

**Enter Dates**

## Instructor Information

| Instructor | Email | Office Location & Hours |
|---|---|---|
| add text | add text | add text |

## General Information

### Description

This course introduces learners to the Python language and is mapped to Certiport's certification exam objectives, a globally accepted, standard-based credential for validating skills.

Learners will begin with a basic introduction to some fundamental programming concepts and the python language, followed by different data structures and operations used in Python.

Control flow using branching and loops are discussed along with an introduction to input and output operations from files and the console. Finally, the topic of modules and packages will be covered. Learners will discover how to code using Python and solve different problems.

Successful completion of the certification exam validates the knowledge and skill sets of individuals seeking employment or advancement in their careers.

### Expectations and Goals

Upon completion of this course, learners are expected to complete the Python IT Specialist Certification Exam. Candidates for this exam will demonstrate that they can recognize, write, and debug Python code that will logically solve a problem. To learn more about the Information Technology Specialist program visit: IT Specialist Certifications: Certiport (pearsonvue.com) Certifications provide significant advantages to professional and job candidates. These include:

- Higher grade point average for certified high school students
- Higher graduation rates for certified high school students
- Increased post-secondary enrollment
- Reduced dropout rates

Additional information: The value of certification

## Course Materials

### Required materials

• XperienceED account

• Computing Device

• Internet Connection

### Optional materials

• Headset

# Python

## Level 1

### Unit 2

### Python Non-Primitive Data Structures



IT INFORMATION TECHNOLOGY SPECIALIST

| Unit | Lesson | Lesson Topic | Self Study Timings in minutes | Instructor-led | Certiport OD | Certiport Description |
|---|---|---|---|---|---|---|
| **Unit 2: Python Non-Primitive Data Structures** | | | | | | |
| | Lesson 1: Objects and Data Structures | | | | | |
| | | Lesson Objectives | 0 | 5 | | |
| | | Combine Different Data Types | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Python Built-In Objects - Strings, Numbers and | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Variables Mutability | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Structured Built-In Objects | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Lesson Practice Exercise | 10 | 10 | | |
| | | Lesson Practice Questions | 15 | 15 | | |
| | | Total Time to Complete Lesson | 45 | 90 | | |
| | Lesson 2: Lists | | | | | |
| | | Lesson Objectives | 0 | 5 | | |
| | | Define Lists | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Add Items to Lists | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Remove Items from Lists | 5 | 15 | 1.2 | Perform data and data type operations |
| | | List Methods | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Lesson Practice Exercise | 30 | 30 | | |
| | | Lesson Practice Questions | 15 | 15 | | |
| | | Total Time to Complete Lesson | 65 | 110 | | |
| | Lesson 3: Lists Manipulation Techniques | | | | | |
| | | Lesson Objectives | 0 | 5 | | |
| | | List Concatenation | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Nested Lists | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Enumerate() Method in Python | 5 | 15 | 1.2 | Perform data and data type operations |
| | | The copy() Method | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Lesson Practice Exercise | 30 | 30 | | |
| | | Lesson Practice Questions | 15 | 15 | | |
| | | Total Time to Complete Lesson | 65 | 110 | | |
| | Lesson 4: Strings as Lists | | | | | |
| | | Lesson Objectives | 0 | 5 | | |
| | | Strings | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Slicing | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Deletion | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Concatenation | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Iteration | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Membership Check | 5 | 15 | 1.2 | Perform data and data type operations |
| | | String Methods | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Lesson Practice Exercise | 30 | 30 | | |
| | | Lesson Practice Questions | 15 | 15 | | |
| | | Total Time to Complete Lesson | 80 | 155 | | |
| | Lesson 5: Tuples | | | | | |
| | | Lesson Objectives | 0 | 5 | | |
| | | Define and Enumerate Tuples | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Tuple Elements | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Change Tuples | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Concatenate Tuples | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Pack and Unpack a Tuple | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Tuple Methods | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Lesson Practice Exercise | 20 | 20 | | |
| | | Lesson Practice Questions | 15 | 15 | | |
| | | Total Time to Complete Lesson | 65 | 130 | | |
| | Lesson 6: Sets | | | | | |
| | | Lesson Objectives | 0 | 5 | | |
| | | Define Sets | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Add Elements to a Set | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Use Elements | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Remove Elements from a Set | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Iterate Over a Set Using the enumerated for Loop | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Nest Sets | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Set Methods | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Lesson Practice Exercise | 30 | 30 | | |
| | | Lesson Practice Questions | 15 | 15 | | |
| | | Total Time to Complete Lesson | 80 | 155 | | |
| | Lesson 7: Literals | | | | | |
| | | Lesson Objectives | 0 | 5 | | |
| | | Literals | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Numeric Literals | 5 | 15 | 1.2 | Perform data and data type operations |
| | | String Literals | 5 | 15 | | |
| | | None, the Special Literal | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Boolean Literals | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Literal Collections | 5 | 15 | 1.2 | Perform data and data type operations |
| | | Lesson Practice Exercise | 20 | 20 | | |
| | | Lesson Practice Questions | 15 | 15 | | |
| | | Total Time to Complete Lesson | 65 | 130 | | |
| Create Project | | | 40 | 40 | | |
| Objective Assessment | | | 40 | 40 | | |
| Time to Complete Unit | | Minutes | 480 | 850 | | |
| | | Hours | 8.0 | 14.2 | | |

# Learning Plan

## Python Unit 2: Python Non-Primitive Data Structures

| Instructor: | Class: | Duration: 8 – 14 Hours |
|---|---|---|

### Unit Objectives:

In this unit, you will be introduced to Python data structures, their uses, methods, and the basic concepts of how these data structures behave. In addition, you will learn the similarities and differences between them. Upon successful completion of this unit, you should be able to understand the following:

- Objects and Data Structures
- Lists
- List Manipulation Techniques
- Strings as Lists
- Tuples
- Sets
- Literals

### Essential Questions:

1. What are Python data structures, and how do you use them?
2. What are the similarities and differences between Python data structures?

### Learning Targets:

I will understand the uses of data structures in Python.
So I can begin applying their methods and concepts when programming the Python language.
I know I succeeded when I use the correct data structure when programming in Python.

| Methods and Materials: | Formative Assessments: | Summative Assessments: |
|---|---|---|
| • Lectures<br>• Reading<br>• Videos<br>• Hand-on activities<br>• Creating<br>• Analyzing<br>• Discussing<br>• Teaching | • Learn Tasks<br>• Practice Questions<br>• Practice Exercises | • Objective Assessment<br>• Create Project |

| STEAM | Work Readiness | 5 C's | Blooms Level |
|---|---|---|---|
| ☐Science<br>☒Technology<br>☐Engineering<br>☐Art<br>☒Math | ☐Communication<br>☒Problem solving<br>☒Teamwork<br>☒Work ethic<br>☒Empathy<br>☒Conflict resolution<br>☒Active listening<br>☒Time management<br>☒Adaptability<br>☒Reading<br>☒Mathematics | ☐Critical Thinking<br>☒Creativity<br>☒Communication<br>☒Collaboration<br>☒Citizenship | ☐Remembering<br>☒Understanding<br>☒Applying<br>☒Analyzing<br>☒Evaluating<br>☒Creating |

# Learning Activities

| Lesson | Time Allowed | Content |
|---|---|---|
| Objects and Data Structures | 45-90 minutes | • Combine Different Data Types<br>• Python Built-In Objects - Strings, Numbers and Booleans<br>• Variables Mutability<br>• Structured Built-In Objects |
| Lists | 65-110 minutes | • Define Lists<br>• Add Items to Lists<br>• Remove Items from Lists<br>• List Methods |
| List Manipulation Techniques | 65-110 minutes | • List Concatenation<br>• Nested Lists<br>• Enumerate() Method in Python<br>• The copy() Method |
| Strings as Lists | 80-155 minutes | • Strings<br>• Slicing<br>• Deletion<br>• Concatenation<br>• Iteration<br>• Membership Check<br>• String Methods |
| Tuples | 65-130 minutes | • Define and Enumerate Tuples<br>• Tuple Elements<br>• Change Tuples<br>• Concatenate Tuples<br>• Pack and Unpack a Tuple<br>• Tuple Methods |
| Sets | 80-155 minutes | • Define Sets<br>• Add Elements to a Set<br>• Use Set Elements<br>• Remove Elements from a Set<br>• Iterate over a Set Using Enumerated for Loop<br>• Nest Sets<br>• Set Methods |
| Literals | 65-130 minutes | • Literals<br>• Numeric Literals<br>• String Literals<br>• None, the Special Literal<br>• Boolean Literals<br>• Literal Collections |

# Extension Activities

1. Add additional information to the course portfolio. Portfolios should include evidence of work, reflect on learned skills and how you can incorporate the skills in a current or future project. This is an ongoing extension activity. Continue to add to the portfolio throughout the course.
2. Select one topic learned throughout the unit, then create an instructional video, tutorial, lecture, or hands-on activity to teach others about the skill.
3. Collect data from a science class, then organize the data using Python data structure methods.
4. Research soft skills needed to become a computer programmer. List five soft skills, then define the skills and include an example of the skill in use.

# Python Unit 2 Key Terms

| Term | Definition |
|---|---|
| **Objects** | Data in Python is represented by logical containers, called *objects*. In a Python program, all data is stored as objects or as relationships between objects. Data variables such as integers, strings, and floats are treated as objects in Python. |
| **Class** | Is a blueprint for making a new object. It is considered the outline that describes an object. During the execution of a program, an instance of a class is created as an object following the specifications of such class. Every data type is represented by a different class. The class defines the different methods to be used with this data type and what are their specific purpose. |
| **Mutability** | Can describe the behavior of variables when we assign them to other variables. When a variable is immutable, once a value is linked to a name, that value can't be changed. |
| **Lists** | Store groups of items in an ordered manner. The order of the items in the list will not change. If you add an item to the list, this item will be appended at the end of the list. |
| **Tuple** | Is a group of items stored in an ordered manner. However, Tuples are immutable, while lists are mutable. Which means we cannot modify Tuples after creating them. Tuples allows duplication as well. |
| **Set** | Unlike Lists and Tuples, a *Set* is a group of items that are not ordered, and these items are unique as they cannot be repeated. |
| **Concatenation** | Is combining two or more variables into a single variable. |
| **List Insert () method** | Inserts an item to a specific index position in a list. |
| **List append() method** | Adds elements at the end of the list. |
| **List extend() method** | Adds two lists to each other. |
| **List remove() method** | Removes an item from a list. |
| **List pop() method** | Removes items using their indexes. |
| **List del() method** | Is like pop(). However, it removes items within a defined index range. |
| **List clear() method** | Is used to empty a list from all its items. |
| **List index() method** | Is used to search for a specific element in the list, determines its index, and returns it. |
| **List count() method** | Returns how many occurrences of an element exists in a certain list. |
| **List reverse() method** | Reorders the list elements in the opposite way. |
| **List sort() method** | Is a particularly important method, and it can be used in several ways. |
| **List copy() method** | Makes a new list that stores the same values of another. |
| **String** | Is a sequence of characters, and we can access these characters individually or by range. In most cases, strings are used like lists. |

# Unit 2: Unplugged Activities Instructor Guide

## Instructions

Below are a variety of offline activities to choose from to support learning in Unit 2. Choose activities to enhance learning in the classroom.

### Activity 1: Word Search

Distribute pages 2-4 to the learners.

**With Words** - Use this word search to reinforce the key terms in Unit 2. An answer key is provided.

**With Clues** - Use this word search to challenge learners to find key terms in Unit 2 using clues. An answer key is provided.

### Activity 2: Crossword Puzzle

Distribute pages 2-3 to the learners.

Have learners solve the crossword puzzle by reading clues and filling in the answer with key terms from Unit 2.

An answer key is provided.

### Activity 3: Cross-out-3 Game

In this activity, the learners will play a game of cross-out-3.

Distribute the activity sheet on page 3 to the learners.

Provide learners with the list of terms by writing them on the board, projecting them or providing a printed list.

Ask the learners to select 9 words from the list of terms and fill their grid with the words.

You will then select from the terms randomly and read only their definition. Then the learners must guess the term in order to cross it off their grid if present.

Continue reading the definitions until one learner crosses out 3 words in a row vertically, horizontally, and diagonally and shouts 'done'.

### Activity 4: Forbidden Words Game

In this activity, the class will play a word game. First, divide the learners into two groups. Print and cut the cards beforehand. The rules of the game are provided in the activity instructions.


1

# Lesson 1: Objects and Data Structures

Unit 2: Python Non-Primitive Data Structures





2

# Combine Different Data Types

**Code**

```
var_S = 3.2
var_F = 4.5
print(type(var_S))
print(type(var_F))
print(var_S + var_F)
```

**Output**

- <class 'float'>
- <class 'float'>
- 7.7





3

# Combine Different Data Types

**Code**

```
var_S = "3.2"
var_F = 4.5
print(type(var_S))
print(type(var_F))
print(float(var_S)+ var_F)
```

**Output**

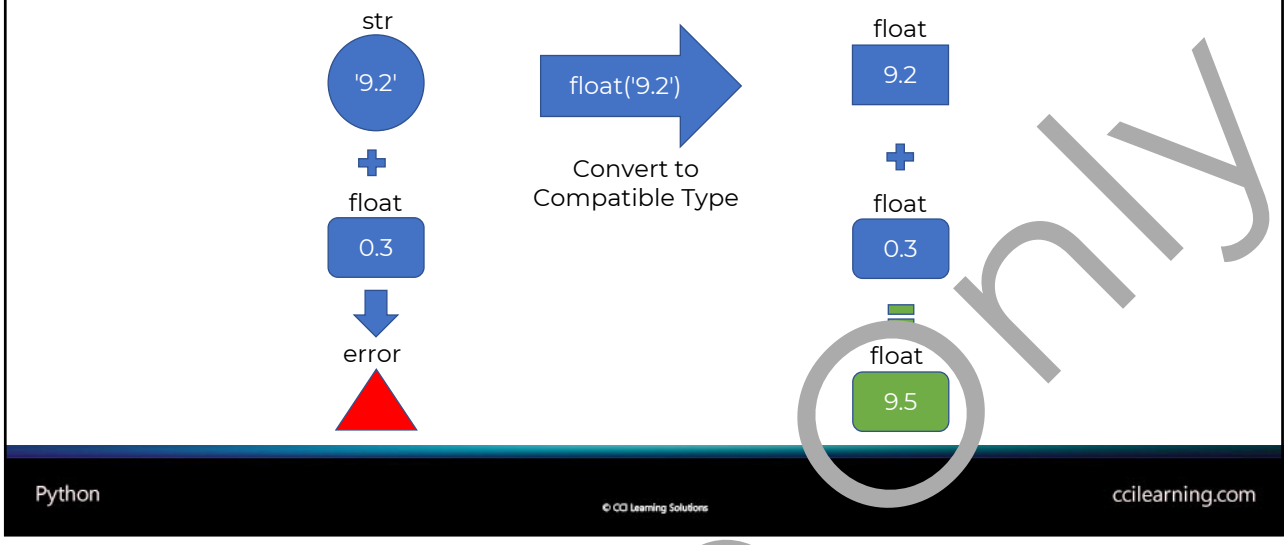
- <class 'str'>
- <class 'float'>
- 7.7





4

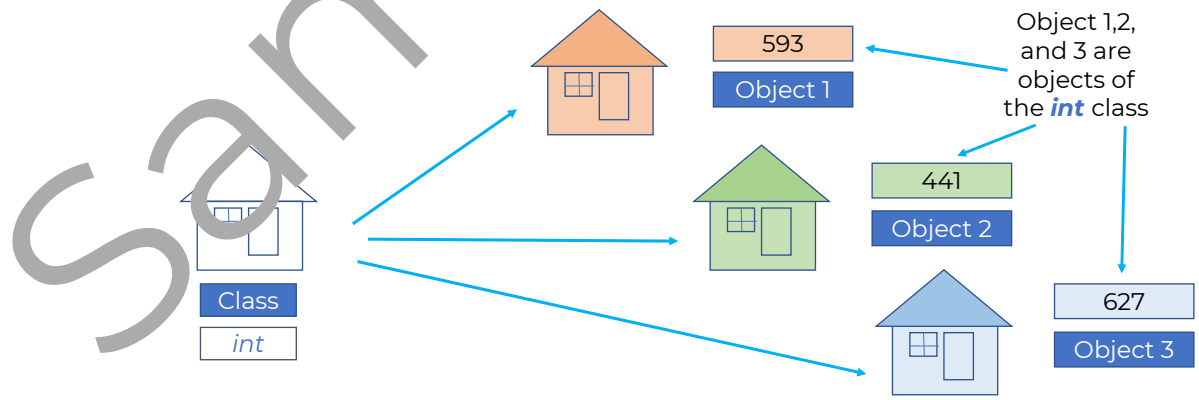
# Combine Different Data Types






5

# Python Built-In Objects




***Class*** is a blueprint for making a new object.



6

# Unit 2 Lesson 1 Study Guide Complete

<table>
<tr><th>Topic</th><th>Explanation</th></tr>
<tr><td>Combine Different Data Types</td><td>
Each Python data type has properties. These properties are different for each data type. You can combine data types that are of the same type. For example, you can combine a float variable with another float variable. You cannot combine a string with a float.<br>
In the following example, we will combine two float variables.<br>
<code>var_S = 3.2</code><br>
<code>var_F = 4.5</code><br>
<code>print(type(var_S))</code><br>
<code>print(type(var_F))</code><br>
<code>print(var_S + var_F)</code><br>
The following is a line-by-line explanation of the above code:
<table>
<tr><th>Code Lines</th><th>Description</th></tr>
<tr><td>var_S = 3.2</td><td>Assigns var_S the value of 3.2.</td></tr>
<tr><td>var_F = 4.5</td><td>Assigns var_F the value of 4.5.</td></tr>
<tr><td>print(type(var_S))</td><td>Displays the data type of variable var_S which is float.</td></tr>
<tr><td>print(type(var_F))</td><td>Displays the data type of variable var_F which is float.</td></tr>
<tr><td>print(var_S + var_F)</td><td>Adds the value of var_S and the value of var_F and displays the addition result.</td></tr>
</table>
The following figure illustrates the above code and its output:<br>


Figure 1-1: Combining two float variables<br>
To combine different data types, you need to convert one so that both of them are of the same type. You can convert a string variable to be a float. For example, <code>var_S</code> is a string variable, you can use <code>float(var_S)</code> to convert it to a float variable. Then, you can combine it to <code>var_F</code>.<br>
<code>var_S = "3.2"</code><br>
<code>var_F = 4.5</code><br>
<code>print(type(var_S))</code>
</td></tr>
</table>

# Unit 2 Lesson 1 Learn Tasks

| Task Level | Certiport OD | Certiport Description | Lesson Topic | Assessment Type | Assessment Details | Input (Answer Code) | Output or Answer Key |
|---|---|---|---|---|---|---|---|
| 2 | 1.2 | Perform data and data type operations | Combine Different Data Types | Code-Check | var_S is a String variable with a value of = "3.2", use float(var_S) to convert it to a float variable, adding to it the value 4.5, print the result and the type of the different variable | var_s="3.2"<br>var_float=float (var_s)+4.5<br>print(var_float )<br>print(type(var_s))<br>print(type(var_float)) | 7.7<br><class 'str'><br><class 'float'> |
| 2 | 1.2 | Perform data and data type operations | Python Built-In Objects - Strings, Numbers and Booleans | Code-Check | Use the function instance to check the type of the object x against the appropriate data type, where x = 1 | x=1<br>print(isinstance( x,int)) | True |
| 4 | 1.2 | Perform data and data type operations | Variables Mutability | Question Set | 1. True/False<br>When a variable is immutable, and a value linked to this named variable, that value can be changed.<br>2. Essay | | 1. True<br>**False (correct)**<br>**2. Keywords**<br>**List**<br>**Sets**<br>**Dictionaries**<br>**Sample Solution:** |

# Unit 2 Lesson 1 Practice Questions

| Assessment Type | Assessment Details | Answer Key |
|---|---|---|
| Multiple Choice | What is the expected output of this code?<br>`var_int = 99`<br>`var_sum = 100`<br>`print("Sum of " + str(var_int) + " + 1 = " + str(var_sum))`<br>• `Sum of 99 + 1 = 100`<br>• `Sum of 100 + 1 = 99`<br>• `Sum of 99 + 100 = 1` | • **`Sum of 99 + 1 = 100` (correct)**<br>• `Sum of 100 + 1 = 99`<br>• `Sum of 99 + 100 = 1` |
| Fill in the Blanks | Fill in the blank with the correct word.<br>A Python ***_*** is a blueprint for making a new object. It is considered the outline that describes an object. | A Python ***Class* (correct)** is a blueprint for making a new object. It is considered the outline that describes an object. |
| True/False | The output of the following code is `True`<br>`f = 4.4`<br>`isinstance (x, bool)` | True<br>**False (correct)** |
| Multiple Choice | Select all the Python immutable built-in types (Select all that apply)<br>• Strings<br>• Numbers<br>• Sets<br>• Lists<br>• Booleans<br>• Tuples | • **Strings (correct)**<br>• **Numbers (correct)**<br>• Lists<br>• **Booleans (correct)**<br>• **Tuples (correct)** |
| Multiple Choice | Given this code<br>`li = [3, 4, 5]`<br>Which code will display the type of variable `li`?<br>• `len(li)`<br>• `print(li)`<br>• `print(type(li))` | • `len(li)`<br>• `print(li)`<br>• `print(type(li))` **(correct)**<br>• `type(li)` |

# Unit 2 Lesson 1 Practice Exercises

| Level | Exercise Number | Assessment Details | Input (Correct Code) | Output or Answer Key | File Name |
|---|---|---|---|---|---|
| 2 | 1 | Input the following code, then observe the output.<br>`myVar = 321`<br>`print(isinstance(myVar, int))`<br>`myVar = str(myVar)`<br>`print(isinstance(myVar, int))`<br>`days = ["Sunday", "Monday", "Tuesday"]`<br>`print(type(days))`<br>`other_days = days`<br>`other_days.append("Wednesday")`<br>`print(days)` | `myVar = 321`<br>`print(isinstance(myVar, int))`<br>`myVar = str(myVar)`<br>`print(isinstance(myVar, int))`<br>`days = ["Sunday", "Monday", "Tuesday"]`<br>`print(type(days))`<br>`other_days = days`<br>`other_days.append("Wednesday")`<br>`print(days)` | `True`<br>`False`<br>`<class 'list'>`<br>`['Sunday', 'Monday', 'Tuesday', 'Wednesday']` | |