

Python Programming Quiz - Lecture 1: Solution Key

Course: Introduction to Python Programming

Instructions

This document provides the solutions for the Python Programming Quiz - Lecture 1. It includes answers for all multiple-choice questions (1 mark each) and complete solutions for programming questions (4 marks each).

1 Multiple-Choice Questions (10 marks)

1.1 Question 1: Introduction to Python

Question: Who developed the Python programming language?

- (a) Dennis Ritchie (b) Guido van Rossum (c) James Gosling (d) Bjarne Stroustrup

Answer: (b) Guido van Rossum

Explanation: As per the lecture slides, Python was developed by Guido van Rossum.

1.2 Question 2: Introduction to Python

Question: Which feature of Python makes it platform-independent?

- (a) Simple & Easy (b) Free & Open Source (c) Portable (d) High-Level Language

Answer: (c) Portable

Explanation: The slides state that Python is portable, meaning it can run on various platforms without modification.

1.3 Question 3: Python Character Set

Question: Which of the following is NOT part of Python's character set?

- (a) Letters (A-Z, a-z) (b) Digits (0-9) (c) Reserved Keywords (d) Special Symbols (+, -, *, /)

Answer: (c) Reserved Keywords

Explanation: The slides list letters, digits, special symbols, and whitespaces as part of Python's character set. Reserved keywords are not included in the character set.

1.4 Question 4: Variables and Identifiers

Question: Which of the following is a valid Python identifier?

- (a) 1variable (b) my_{variable} (c) @variable (d) variable#

Answer: (b) my_{variable}

Explanation: According to the slides, valid identifiers can include letters, digits, and underscores, but cannot start with a digit or contain special characters like @ or #.

1.5 Question 5: Variables and Identifiers

Question: Which symbol is NOT allowed in a Python identifier?

- (a) Underscore (b) Dollar Sign (\$) (c) Letters (a - z) (d) Digits (0 - 9)

Answer: (b) *DollarSign(\$)*

Explanation: The slides specify that special symbols like \$ are not allowed in identifiers, while underscore is allowed.

1.6 Question 6: Data Types and Keywords

Question: Which of the following is NOT a valid Python data type?

- (a) Float (b) Boolean (c) Integer (d) Double

Answer: (d) Double

Explanation: The slides list Integer, Float, Boolean, String, and None as valid data types. Double is not a Python data type.

1.7 Question 7: Comments

Question: How is a single-line comment written in Python?

- (a) // Comment (b) # Comment (c) /* Comment */ (d) <!-- Comment -->

Answer: (b) # Comment

Explanation: The slides indicate that single-line comments in Python are written using the # symbol.

1.8 Question 8: Operators

Question: Which operator performs division in Python?

- (a) * (b) / (c) % (d) //

Answer: (b) /

Explanation: The slides mention division as a special symbol (/), which performs floating-point division in Python.

1.9 Question 9: Type Conversion and Type Casting

Question: What is the output of the following code?

```
a, b = 4, "6"
```

```
c = int(b)
```

```
print(a + c)
```

- (a) 10 (b) 46 (c) "10" (d) Error

Answer: (a) 10

Explanation: The string "6" is converted to an integer using `int(b)`, and `4 + 6` equals 10, as shown in the slides type casting examples.

1.10 Question 10: Input in Python

Question: What is the default data type of the value returned by the `input()` function?

- (a) Integer (b) Float (c) String (d) Boolean

Answer: (c) String

Explanation: The slides state that the `input()` function always returns a string.

2 Programming Questions (20 marks)

2.1 Question 11: Type Conversion

Question: Write a Python program that converts a string "12.5" to a float, adds it to an integer 8, and prints the result.

Solution:

```
a = "12.5"
b = float(a)
c = 8
result = b + c
print(result) % Output: 20.5
```

Explanation: The string "12.5" is converted to a float using `float()`, then added to the integer 8, as covered in the type casting section of the slides.

2.2 Question 12: Input and String Concatenation

Question: Write a Python program that accepts a string input from the user and concatenates it with "Hello, " to print a greeting message.

Solution:

```
name = input("Enter your name: ")
greeting = "Hello, " + name
print(greeting)
```

Explanation: The `input()` function captures a string, which is concatenated with "Hello, " using the `+` operator, as implied in the slides input and programming exercises.

2.3 Question 13: Basic Programming

Question: Write a Python program to input two integers and print their sum.

Solution:

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
sum = a + b
print(sum)
```

Explanation: The program uses `input()` to get two strings, converts them to integers using `int()`, and prints their sum, as shown in the slides "Print Sum" example.

2.4 Question 14: Basic Programming with String Concatenation

Question: Write a Python program to input the side of a square, calculate its area, and print the result with a concatenated string in the format "Area: <area>".

Solution:

```
side = float(input("Enter the side of the square: "))
area = side * side
print("Area: " + str(area))
```

Explanation: The program takes a float input for the side, calculates the area (side * side), and concatenates the result with "Area: " after converting the area to a string, as per the slides exercises.

2.5 Question 15: Basic Programming

Question: Write a Python program to input two integers, **a** and **b**, and print **True** if **a** is greater than or equal to **b**, otherwise print **False**.

Solution:

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print(a >= b)
```

Explanation: The program takes two integer inputs, compares them using the **>=** operator, and prints **True** or **False**, as specified in the slides programming exercises.