






CODETANTRA

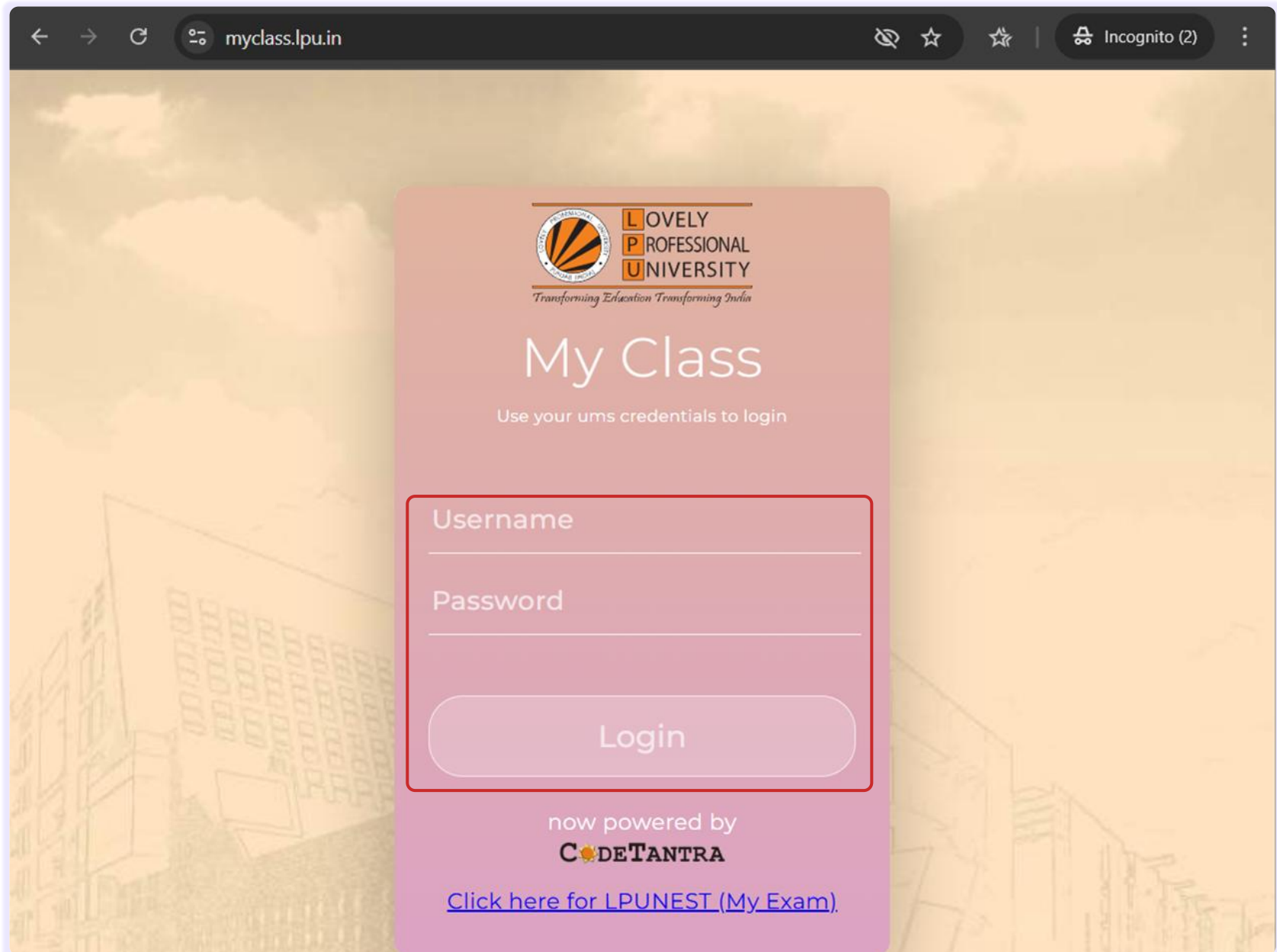
Coding Platform

A comprehensive guide for students



Supported Operating Systems & Browsers

Operating System	 Windows	 Mac	 Linux
Supported Browser	 CodeTantra Secure Exam Application (SEA)	 CodeTantra Secure Exam Application (SEA)	 CodeTantra Secure Exam Application (SEA)



1. Login to myclass.lpu.in
2. Enter your UMS credentials to log in to the portal

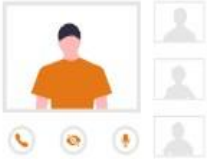
CodeTANTRA Teach & Learn Anywhere

lovelyprofessionaluniversity.codetantra.com/secure/home.jsp

Incognito (2)


CODETANTRA Home Learn Anywhere

steve@gmail.com Support Logout




Click here to join or view all your live and recorded classes/meetings

View Classes/Meetings




Click here to view all your courses/subjects

Courses



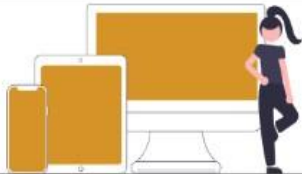
Click here to view all your scheduled and completed tests

Tests




Click here to view all your programming labs

Programming Labs



Click here to access tools.

Tools



Click here to reach us

Help & Support

Go to Courses

A. Click on Courses card available on the home page.

COURSES

[← Prev](#) [Next →](#)

Less More

[Enrolled Courses](#)

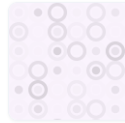
3

**Python Programming - Aug 2024 - INT108@24251** v2

This course covers all the concepts of Python Programming Language as a Lecture schedule-wise

**C Programming - Lesson Plan**

This course covers all the concepts of C programming language as a Lecture schedule-wise

**Java Programming - 2023**

Java Programming - 2023

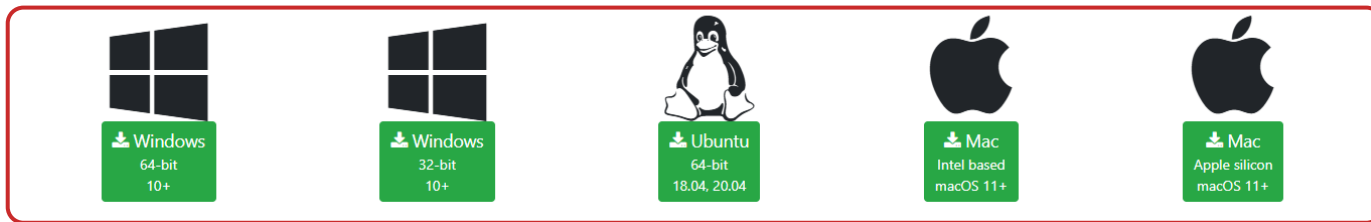


Select the course from the list of enrolled courses.

This course can only be taken in CodeTantra Secure Exam App.

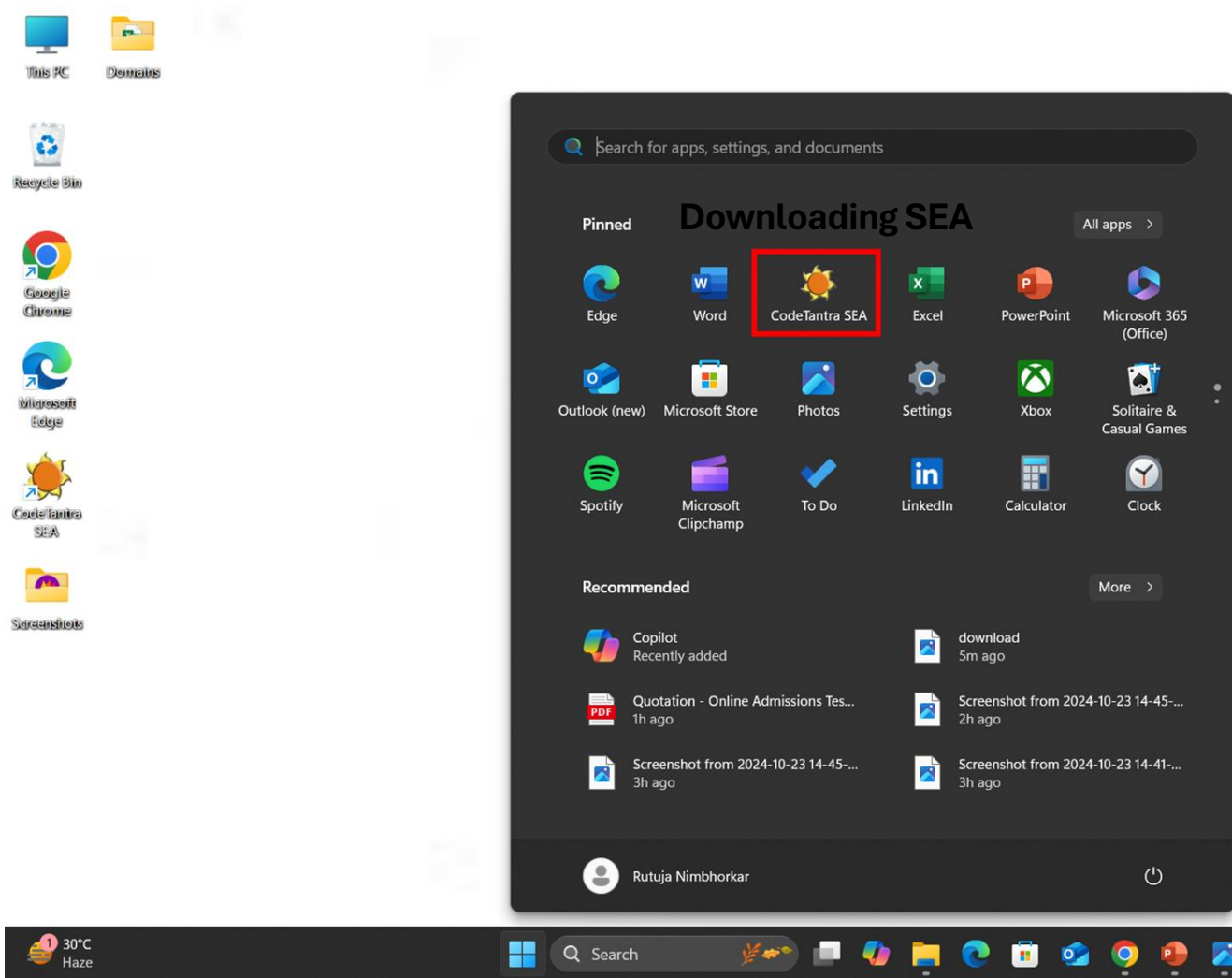
If you already have **CodeTantra Secure Exam App** installed on your computer, please login using that to take the course.

Click on Download Now to get the latest version of **CodeTantra Secure Exam App**



Click [here](#) to check our [Terms of Use & Privacy Policy](#)

Download the application according to your system configuration.



Run The SEA app in Primary Device

- Go to Start Menu.
- Find and click on CodeTantra SEA app.

Accessing Courses

Codetantra's LMS V2



1



Select your institution

lovelyprofessionaluniversity.codetantra.com



Lovely Professional University

[lovelyprofessionaluniversity.codetantra.com]

2



My Class

Use your ums credentials to login

Username

Password

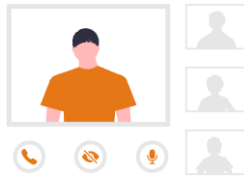
Login

now powered by
CODETANTRA

[Click here for LPUNEST \(My Exam\).](#)

Open the CodeTantra SEA on your Desktop or Laptop running Windows/Mac/ Linux OS.

1. Type **lovelyprofessionaluniversity.codetantra.com** in the Select your institution dropdown search box.
2. Enter your UMS credentials to log in to the portal



Click here to join or view all your live and recorded classes/meetings

[View Classes/Meetings](#)



Click here to view all your courses/subjects

[Courses](#)



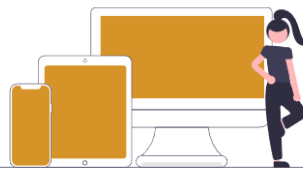
Click here to view all your scheduled and completed tests

[Tests](#)



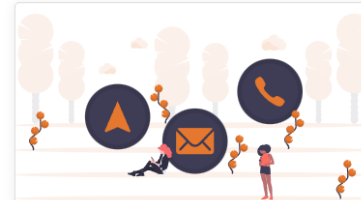
Click here to view all your programming labs

[Programming Labs](#)



Click here to access tools.

[Tools](#)



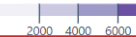
Click here to reach us

[Help & Support](#)

Go to Courses

A. Click on Courses card available on the home page.

COURSES

[← Prev](#) [Next →](#)Less  More[Enrolled Courses](#)

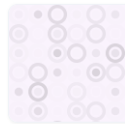
3

**Python Programming - Aug 2024 - INT108@24251** v2

This course covers all the concepts of Python Programming Language as a Lecture schedule-wise

**C Programming - Lesson Plan**

This course covers all the concepts of C programming language as a Lecture schedule-wise

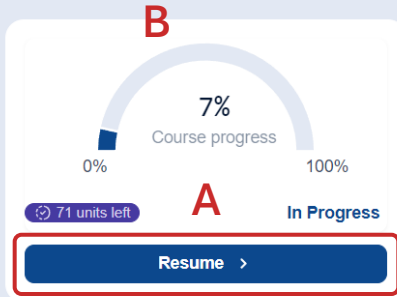
**Java Programming - 2023**

Java Programming - 2023



Select the course from the list of enrolled courses.

Python Programming - Aug 2024 - INT108@24251

[Dashboard](#) [Contents](#) [Calendar](#) [Assessments](#) [Syllabus](#)

Description

This course covers all the concepts of Python Programming Language as a Lecture schedule-wise

Upcoming tests

No upcoming exams in the next 7 days

Instructors

Credits

Course Outcomes

Start/Resume the Course

- Click on the Resume button to continue studying after the last solved question.
- If you are just starting this course, click on the Contents button.

Python Programming - Aug 2024 - INT108@24251

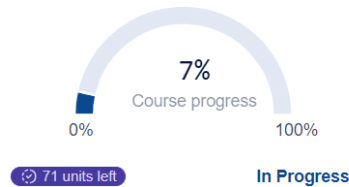
[Dashboard](#) [Contents](#) [Calendar](#) [Assessments](#) [Syllabus](#)

Search course

ctrl + k

A

- 1. Unit 1 - Lesson 1 - Python - Language Fea... ▾
- 2. Unit 1 - Lesson 2 - Comments, Identifiers ... ▾
- 3. Unit 1 - Lesson 3 - Variables and Assignm... ▾
- 4. Unit 1 - Lesson 4 - Expressions and State... ▾
- 5. Unit 1 - Lesson 5 - Data Types - Numbers,... ▾
- 6. Unit 1 - Lesson 6 - Data Types - Lists, Set... ▾
- 7. Unit 1 - Lesson 7 - Data Types - Dictionari... ▾
- 8. Unit 1 - Lesson 8 - Input and Output State... ▾
- 9. Unit 1 - Lesson 9 - Arithmetic Operators, C... ▾



Pickup from where you left off

Introduction to Data Definition Language (DDL)

Resume

L1

Unit • 100% completed

L2

Unit • 100% completed

L3

Unit • 100% completed

L4

Unit • 100% completed

Start/Resume the Course

A. Click on the Question to go to the respective content.

B. Alternatively, You can click on the Resume button also to resume from where you left off.

7.2.1. Data Type Conversion

13:14

Converting data of one type to data of another type is called **Type Conversion**.

Python defines various **type conversion functions** to do this.

1. **int(x, base)**: This function converts **x** to an integer of specified base. If base is not specified, it defaults to **10**.

The syntax of **int()** function is:

```
int(x=0, base=10)
```

- x - Number or string to be converted to integer. Default argument is zero.
- base - Base of the number x can range from **0 (code literal)** or **2** to **36**.

Consider the following program to understand the working of **int()** function.

```
s = "0011" # A binary string.  
print(int(s, 2)) # Converts string type to int type using int() with base 2  
print(int(s)) # base not specified, it defaults to 10
```

Output:
3
11

If "0011" is considered with base 2, then its integer value will be **3**, whereas if "0011" is considered with base 10, then its integer value will be **11**.

2. **float(x)**: This function is used to convert any data type to a floating point number. The float() function returns a floating point number from a number or a string.

Sample Test Cases

Data Type...

Submit

```
1 a = int(input("Enter a value: "))  
2 b = int(input("Enter b value: "))  
3 # print a value respective string  
4 # print a value respective char  
5 # print a value respective hexadecimal value  
6 # print a and b of complex number
```

Code with red background indicates non-editable code

Terminal Test cases

< Prev Reset Submit

Interface Introduction

A. Learning Content Panel: Learn the given content

B. Question Panel: Solve the given problem (It can be coding or MCQ type)

C. Sample Test Cases: Expand the Sample Test Cases panel to know the desired output.

8.1.1. Reading input in Python - strings

01:55

If you want to take only **specified data type** input from the user, you need to mention the data type before the **input**.

For example : `a = int(input("Enter an integer :"))`

So that, if the user gives an input other than the mentioned data type, it results in the `valueError`.

Write a program to print your favourite place.

Sample Test Cases

Test case 1

Enter your favourite place: Mexico

My favourite place is: Mexico

Test case 2

Enter your favourite place: Sydney

My favourite place is: Sydney

Test case 3

Enter your favourite place: Delhi

My favourite place is: Delhi

Input1.py

```
1 place = input("Enter your favourite place: ") #take your favourite place  
   using input statement  
2  
3 #Print your favourite place  
4 print("My favourite place is:", place)
```

Average time

0.006 s
5.60 ms

Maximum time

0.009 s
9.00 ms

3 out of 3 shown test case(s) passed

2 out of 2 hidden test case(s) passed

✓ Test case 1 9 ms

Debug

Expected output

Enter your favourite place: Mexico

My favourite place is: Mexico

Actual output

Enter your favourite place: Mexico

My favourite place is: Mexico

✓ Test case 2 5 ms

✓ Test case 3 4 ms

Terminal

Test cases

< Prev Reset Submit Next >

Solve a Coding Question

A. Coding Problem: Solve the problem by writing the code.

B. Submit: Click on the submit button to submit the code for auto-evaluation

C. Test Cases: This panel provides comparison of the *Expected output* and *Actual output*. These should match 100% character by character in the desired order. A problem will be considered solved only if all *shown* and *hidden* test cases are passed.

6.1.1. Understanding List Creation

03:19

Working with nested lists: The items of the list can be lists themselves, which means that the lists can be nested.

Let us consider example:

```
list1 = [23, 5.65, ["A", 34.23], "India"]
```

In the above example, the third element (i.e **index = 2**) of list1 is a **list**.

We access the **3rd item** by list1[2], which is a list and the second item of this list can be accessed using **index 1**. So, **list1[2][1]** will be **34.23**.

Write the missing code given to understand **List Concatenation**. Follow the instructions given as comment lines in the program.

Sample Input and Output:

```
List1 Elements are: [1.0, 2.3, 'hello']
List2 Elements are: ['hi', 8.3, 9.6, 'how']
List after Concatenation: [1.0, 2.3, 'hello', 'hi', 8.3, 9.6, 'how']
```

Sample Test Cases**Test case 1**

```
List1 Elements are: [1.0, 2.3, 'hello']
List2 Elements are: ['hi', 8.3, 9.6, 'how']
List after Concatenation: [1.0, 2.3, 'hello', 'hi', 8.3, 9.6, 'how']
```

List1.py

```
1 list1 = [1.0, 2.3, "hello"]
2 list2 = ["hi", 8.3, 9.6, "how"]
3
4 #print the list1 and list2
5 print("List1 Elements are:", list2)
6 print("List2 Elements are:", list1)
7
8 #Concatenate list1 and list2 and print the result
9 print("List after Concatenation:", list1+list2)
10
11
```

Average time
0.007 s
7.00 ms

Maximum time
0.007 s
7.00 ms

0 out of 1 shown test case(s) passed

Test case 1

Output mismatch

Expected output	Actual output
List1 Elements are: [1.0, 2.3, 'hello']	List1 Elements are: ['hi', 8.3, 9.6, 'how']
List2 Elements are: ['hi', 8.3, 9.6, 'how']	List2 Elements are: [1.0, 2.3, 'hello']
List after Concatenation: [1.0, 2.3, 'hello', 'hi', 8.3, 9.6, 'how']	List after Concatenation: [1.0, 2.3, 'hello', 'hi', 8.3, 9.6, 'how']

Terminal

Test cases

< Prev

Reset

Submit

Next >

Output Comparison and Code Correction

A. Output Comparison: There are three output statements in Test Case 1. The mismatched character in the actual output has been highlighted in the first output statement using a red rectangle. The student should investigate the code to know why this is happening and make the required corrections in the code.

B. The test case failed because of the wrong order of list1 and list2 printing. Fixing this should resolve the problem.

More on Test Cases

What is a test case?

A test case is a specific combination of input and expected output used to validate the functionality of a program.

How are test cases used on the CodeTantra platform?

Test cases on the CodeTantra platform are employed to automatically evaluate programs written by students. For each test case, the platform inputs data into the student's program, and the program's output is compared to the expected output on a character-by-character basis.

What constitutes a passed or failed test case?

A test case is considered passed if the program's actual output matches the expected output 100% for the given input. Otherwise, the test case is deemed to have failed.

Is output comparison case-sensitive?

Yes, the comparison is case-sensitive. For example, if the expected output is "CAT," and the program produces "cat," this will be considered a mismatch.

Does extra spacing in output lead to errors?

Yes, the comparison is performed character by character, so any extra leading, trailing, or intermediate spaces will cause the output to be considered incorrect. For instance, "CAT" and " CAT" are not equivalent because the first character differs.

What are the different types of test cases?

There are two main types of test cases:

Shown test cases: These are visible to students and can be used to debug the program to ensure it produces the correct output.

Hidden test cases: These are not shown to students. They are used to assess the correctness of the code, promoting deeper problem-solving skills as students are encouraged to consider all possible edge cases.

What is the weightage of a test case?

A program may include multiple test cases, each assigned a specific weight. The total weightage of all test cases sums to 100%. These weightages are used to partially evaluate the code and assign marks. Typically, shown test cases carry less weight compared to hidden test cases, and weightages need not be uniformly distributed among the test cases.

2.2.2. Understanding Python Keywords

00:20

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Python provides a way to **print** the list of keywords in its current version.

```
import keyword # This statement is used to import the keyword module.
print(keyword.kwlist) # kwlist contains all the keywords of Python
```

The output of the above program is:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif',
```

To check whether a given word is a Python keyword or not, we use a built-in function **iskeyword()**. This function returns a **boolean** value, if the given word is a keyword then it returns **True** as output otherwise returns **False**.

Let us consider a few examples:

```
Program - 1:
import keyword # This will import the keyword module
print(keyword.iskeyword('and')) # Here 'and' is a keyword so it prints True as output
Output: True

Program - 2:
import keyword # This will import the keyword module
print(keyword.iskeyword('python')) # Here 'python' is not a keyword so it prints False as output
Output: False
```

Select all the correct statements.

☒ Python version 3.5 has 33 keywords.

Correct!

☐ true is a valid keyword in Python.

Python is a case sensitive language. In Python **True** is a keyword and not **true**.

☒ The keyword **nonlocal** does not exist in Python 2.

Correct!

☒ Interpreter raises an error when you try to use keyword as a **name** of an entity.

Correct! It shows the output as "Invalid Syntax".

☐ A programmer can easily modify the **keywords**.

Programmers can't modify the **keywords** as they are built into the language.

B**C**[< Prev](#) [Reset](#) [Submit](#) [Next >](#)

Solve a Multiple-Choice Question

A. Question: The question test usually appears at the bottom of the learning content panel.

B. Options: Select all the correct Options.

C. Submit: Click on the submit button to validate your selections. Clicking on submit may explain each option if all selected options are correct.

Understanding Content Tree...

Solved Question

The question marked in green with an icon of a question mark indicates a solved question



Test

The last question marked in green with an icon of a question mark indicates the last solved question



Unlocked Question

The question marked in grey with an icon of a question mark indicates an unlocked question



Locked Content

The content marked in grey with a lock symbol icon indicates locked content. Locked content can be unlocked after solving all previous questions if the course is configured to be completed sequentially.



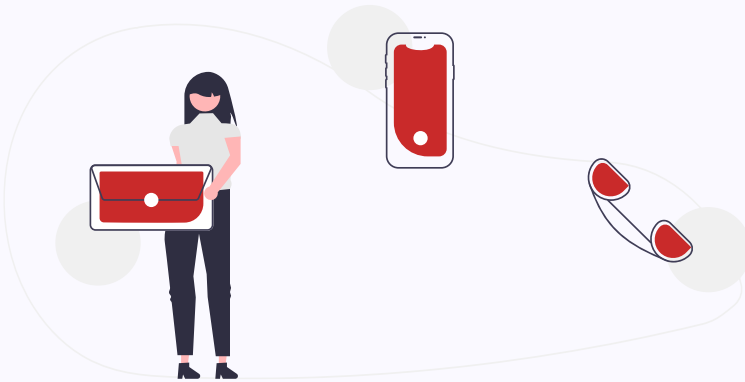
The screenshot displays the CODETANTRA Python Programming Lesson Plan. The left sidebar shows a content tree with items 6. L6, 7. L7, 7.1. Dictionaries, 7.2. Data Type Conversions, 7.2.1. Data Type Conversion, 7.2.2. Data Type Conversions, 7.3. Python Data Types, 8. L8, 8.1. Input and Output Statements, 8.1.1. Reading input in Python - strings, 8.1.2. Understanding Output in Python., 8.1.3. Output of Python, 8.1.4. % - formatting and str.format() function, 8.1.5. % - formatting and str.format() function, 8.2. Basics of Python Programming, 8.3. Python Input/Output Functions, 9. L9, and 10. L10. The main content area shows the 'Input and Output Statements' unit, which includes questions like 'Reading input in Python - strings', 'Understanding Output in Python.', 'Output of Python', and '% - formatting and str.format() function'. The status of each question is indicated by a question mark icon (green for solved, grey for unlocked) and a lock icon (grey for locked).

Thank You

For any technical query,
We are Just a call or mail Away

support@codetantra.com

- +91 799 541 7777
- +91 630 500 7479
- +91 630 500 8997



C☀**DE**TANTRA